MÓDULO SDM 9431

MANUAL PRÁTICO

MÓDULO SDM-9431

MANUAL PRÁTICO

SUMÁRIO

CAPÍTULO 1 - ESPECIFICAÇÕES DO EQUIPAMENTO

1.1 GABINETE

O gabinete é feito de material plástico de alta resistência, apoiando uma placa de circuito impresso, onde estão montados os componentes eletrônicos, um teclado, um display, um conector serial, para comunicação com um microcomputador, e um proto-board, para a realização de montagens experimentais.

Conforme mostram as figuras 1.1 e 1.3, não são usados componentes SMD (de montagem superficial) na placa de circuito impresso superior, o que permite realizar facilmente, se necessárias, manutenções corretivas.

	Largura	365mm
Dimensões Externas	**Profundidade**	310mm
	Altura	307mmm

Figura 1.1 – Módulo SDM-9431.

1.2 PAINEL TRASEIRO

No painel traseiro do SDM-9431 encontram-se:

- Cabo de força (1,5m, condutores 2x0,5mm^2, plug 2 pinos NBR-14136)
- Porta fusível (para fusível 20AG, 1,5A)
- Chave Liga-Desliga
- Chave seletora de tensão (110V/220V)

Figura 1.2 – Painel traseiro do módulo SDM-9431.

1.3 PAINEL SUPERIOR

Figura 1.3 – Vista superior do módulo SDM-9431.

A placa de circuito impresso dupla face que forma o painel frontal contém:

1. **Microcontrolador 8031** (versão sem ROM interna do 8051). Possui uma arquitetura de barramento de dados de 8 bits, instruções de operação de bits, duas fontes de interrupções externas com dois níveis de prioridades programáveis, dois contadores/temporizadores de 16 bits, porta serial com quatro modos de programação e 32 linhas de entrada/saída endereçáveis bit a bit.

2. **Memória RAM externa de 32k bytes,** com opção para uso de memória de 8k bytes, acessada como memória de programa ou como memória de dados, para possibilitar desenvolvimento de programas.

3. **Memória EPROM externa de 16k,** contendo o programa monitor do sistema SDM 9431, que permite a execução de programas em tempo real, no modo passo a passo e no modo "breakpoint". O programa monitor permite que o usuário verifique e altere registros, posições de memória de programa e de dados, bem como que haja comunicação serial com um microcomputador.

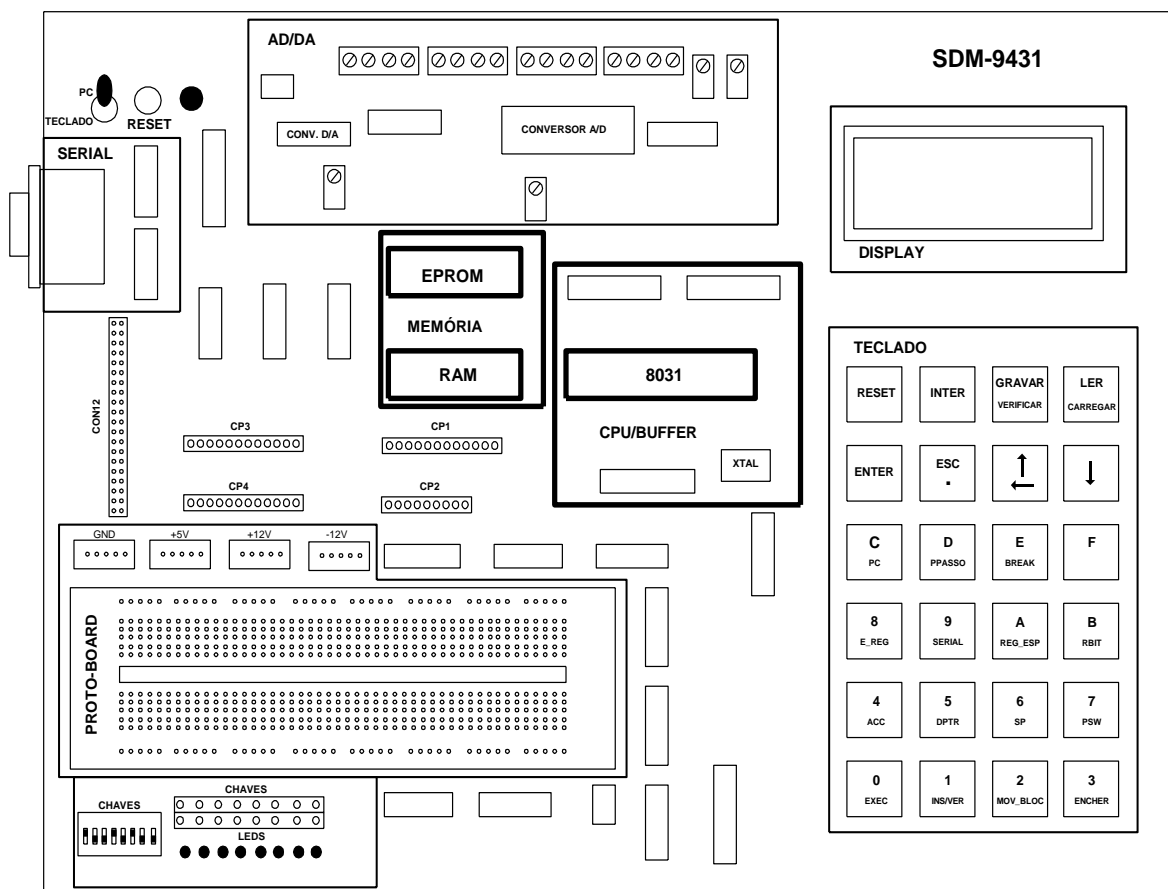



Figura 1.4 – Localização da CPU e das memórias RAM e EPROM .

4. **Teclado de 24 teclas** para comandos e dados hexadecimais, no modo teclado, ou para experiências de entradas de dados, quando operando no modo PC.
5. **Display de cristal líquido alfanumérico**, de duas linhas de 16 caracteres, para comunicação no modo teclado e para saída de dados, quando operando no modo PC.
6. **8 leds e 8 chaves tipo dip switch** para experiências de entrada e saída binária.
7. **Conector serial**, tipo RS-232, para comunicação com microcomputador compatível com IBM-PC.

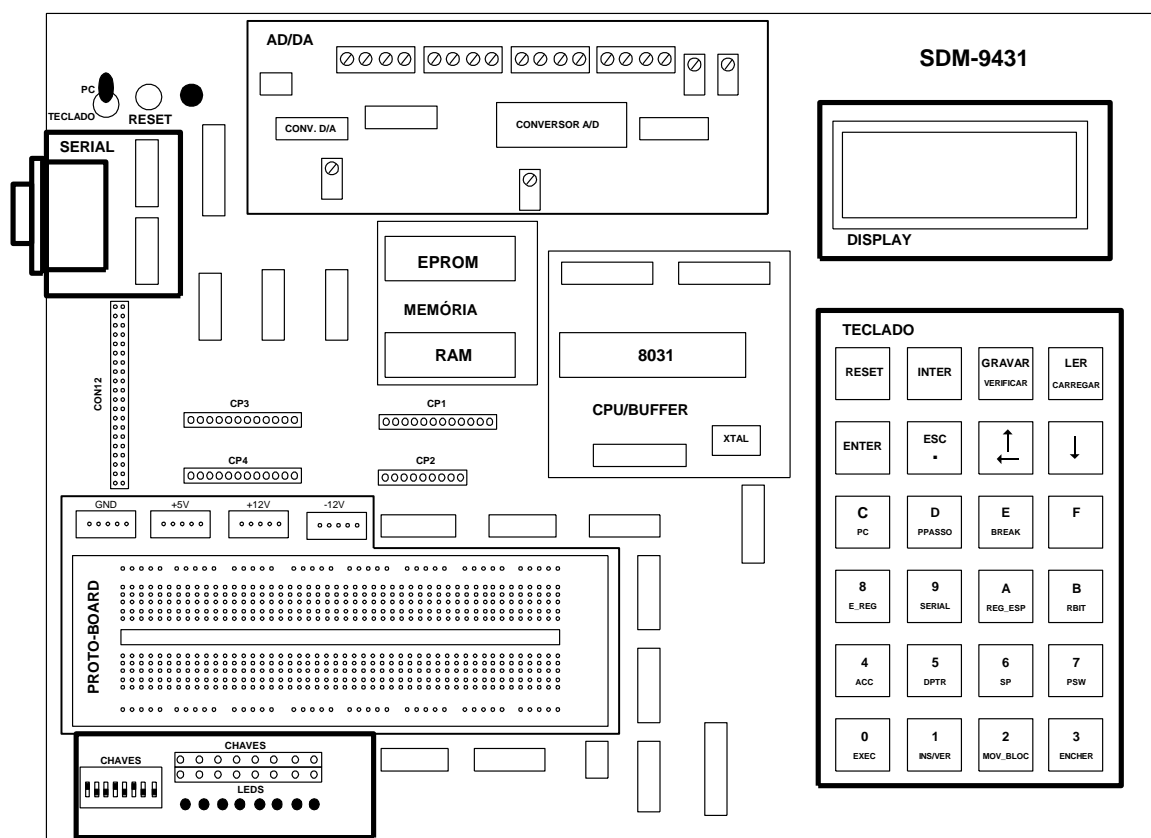



Figura 1.5 – Localização do display, teclado, conector serial, dip-switches e leds.

8. **Conversor análogo-digital** de oito canais multiplexados, de oito bits cada, para implementações de experiências analógicas.
9. **Conversor digital-analógico** de um canal de oito bits para implementações em controle analógico.

10. **Conectores de sinais de barramentos**, amplificados e disponíveis para experiências de análise e desenvolvimento de circuitos de interface, com possibilidade de endereçamentos já decodificados e disponíveis para o usuário.
11. **Proto-board de 550 pontos** para montagem de circuitos experimentais no SDM-9431.
12. **Fontes de alimentação**, junto ao proto-board, possui as seguintes tensões e capacidades de corrente: +5V (3A), +12V (1A) e -12V (1A).

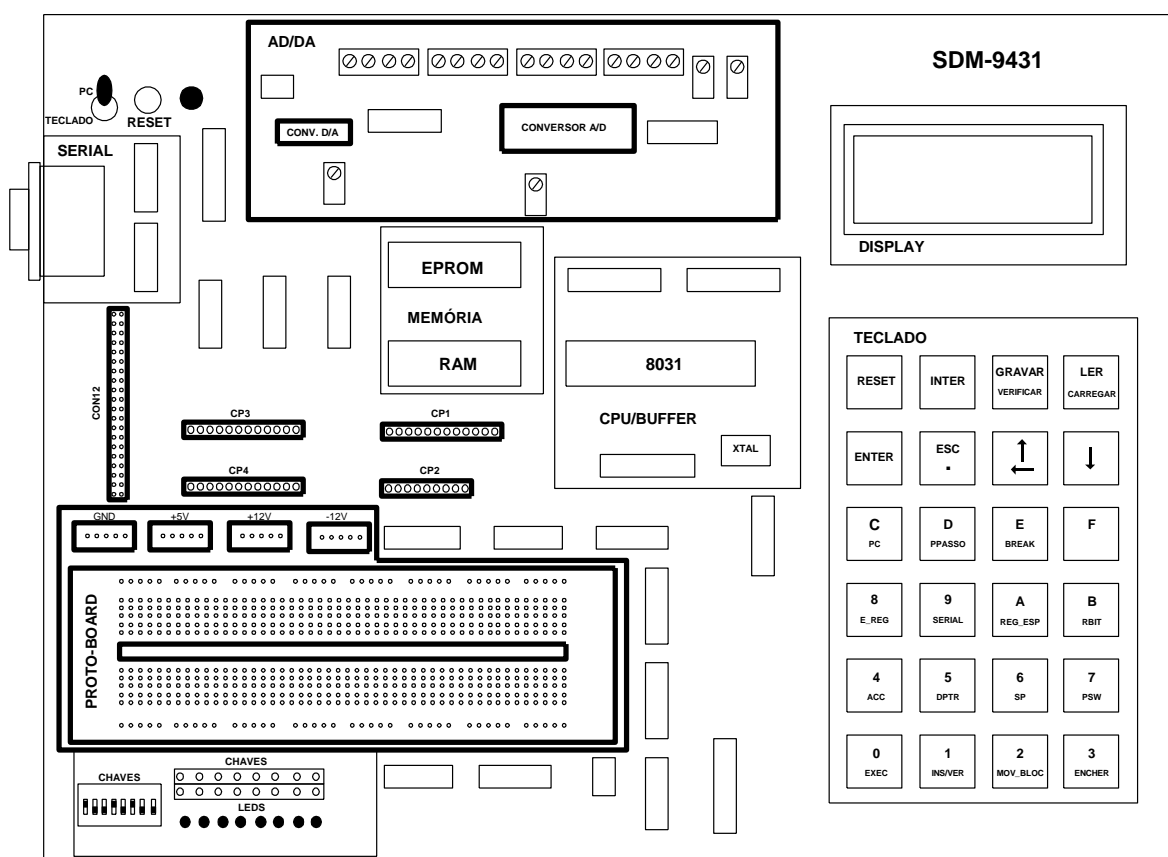



Figura 1.6 – Localização do proto-board, fontes, conversores A/D e D/A e conectores de sinais de barramentos.

CAPÍTULO 2 - OPERAÇÃO NO MODO TECLADO VIA WINDOWS

No ***MODO TECLADO***, o ***Módulo SDM 9431*** terá a sua programação realizada através do seu teclado e do seu display de cristal líquido. Porém, os programas desenvolvidos no sistema poderão ser transferidos para um computador PC, através de comunicação serial, e lidos posteriormente, quando for necessário.

Para iniciar a operação no modo teclado, deve-se selecionar o ***MODO TECLADO***, posicionando a chave de seleção (canto superior esquerdo do equipamento) no modo ***TECLADO*** e ligando o módulo, ou pressionando a tecla ***RESET*** se o módulo já estiver ligado. No display aparecerá a mensagem da figura 2.1.

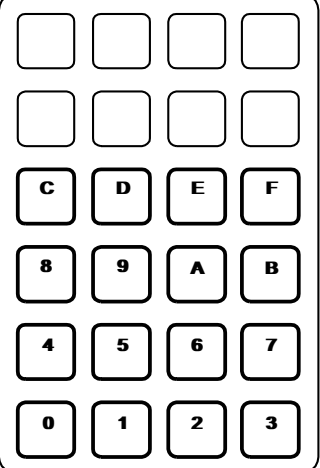



Figura 2.1 - Display inicial para o modo teclado.

O teclado do sistema SDM 9431 é composto por 24 teclas que podem desempenhar mais de uma função. As funções do teclado serão divididas nos seguintes modos: funções de dados, funções auxiliares, funções de registros, funções de comandos e funções do sistema.

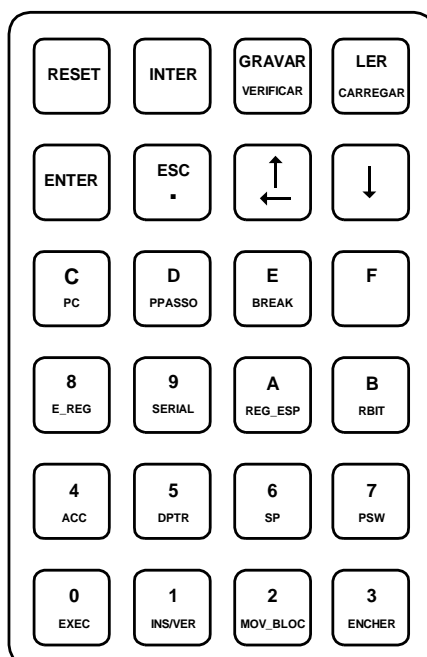
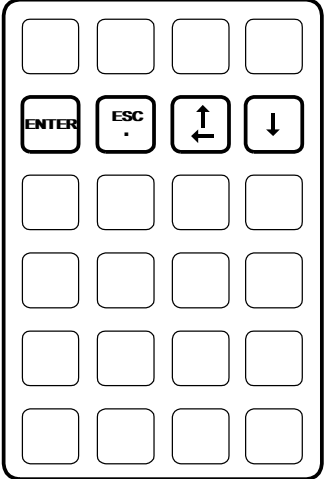
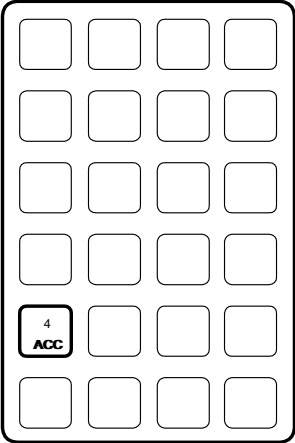
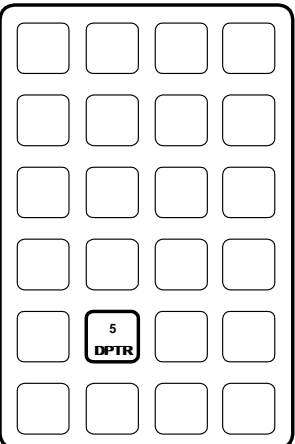
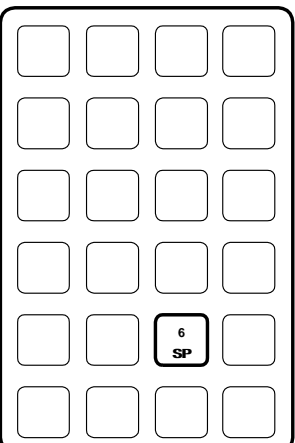



Figura 2.2 – Teclado do módulo SDM-9431.

2.1 FUNÇÕES DE DADOS

Teclas	Descrição
C D E F 8 9 A B 4 5 6 7 0 1 2 3	Usadas para introdução de valores hexadecimais, durante o acesso a um endereço, ou na modificação do conteúdo de um endereço, ou também na modificação do conteúdo de um registro especificado.

2.3 FUNÇÕES AUXILIARES

Teclas		Descrição
ENTER ESC . ↑ ↓		Usadas como funções auxiliares no acesso a endereços ou registros.
	ESC .	Usada para encerrar ou sair de uma função de comando ou função de registro.
	↑ ←	Usada com a função de apagar erros de digitação em todas as funções. Usada para decrementar posições de registros especiais e de memória de dados e de programa, acessados nos comandos ***INS/VER*** e ***REG_ESP.***
	↓	Usada com a função de incrementar endereços nos comandos ***INS/VER*** e ***REG_ESP.***
	ENTER	usada para a confirmação dos endereços ou dados digitados.

2.4 FUNÇÕES DE REGISTROS

As seguintes teclas serão usadas para acessar um registro de função especial do 8031, permitindo a verificação ou alteração dos seus respectivos conteúdos.

Tecla	Descrição
4 ACC	Usada para verificar ou alterar do conteúdo do acumulador. ***Exemplo:*** *Para verificar o conteúdo do acumulador, pressionar a tecla **ACC**. O conteúdo do acumulador será apresentado.* A C C = 0 0 *Para alterar o valor do acumulador, entrar com o novo valor e pressionar a tecla **ENTER**, para confirmá-lo. Pressionar a tecla **ESC** para finalizar a operação.*
5 DPTR	Usada para verificar ou alterar do conteúdo do registro **DPTR**. ***Exemplo:*** *Para verificar o conteúdo do registro, pressionar a tecla **DPTR**. O conteúdo do registro será apresentado.* D P T R = 0 0 0 0 *Para alterar o valor do registro, entrar com o novo valor e pressionar a tecla **ENTER**, para confirmá-lo.Pressionar a tecla **ESC** para finalizar a operação.*
6 SP	Usada para verificar ou alterar o conteúdo do registro apontador de pilha ("stack pointer"). ***Exemplo:*** *Para verificar o conteúdo do registro, pressionar a tecla **SP**. O conteúdo do registro será apresentado.* S P = 0 7 *Para alterar o valor do registro, entrar com o novo valor e pressionar a tecla **ENTER**, para confirmá-lo. Pressionar a tecla **ESC** para finalizar a operação.*

<table>
<tr><td>7
PSW</td><td>Usada para verificar ou alterar o conteúdo do registro de palavra de status do processador.

Exemplo: Para verificar o conteúdo do registro, pressionar a tecla PSW. O conteúdo do registro será apresentado.

P S W = 0 0

Para alterar o valor do registro, entrar com o novo valor e pressionar a tecla ENTER, para confirmá-lo. Pressionar a tecla ESC para finalizar a operação.</td></tr>
<tr><td>C
PC</td><td>Usada para verificar ou alterar o conteúdo do contador de programa.

Exemplo: Para verificar o conteúdo do registro, pressionar a tecla PC. O conteúdo do registro será apresentado.

P C = 0 0 0 0

Para alterar o valor do registro, entrar com o novo valor e pressionar a tecla ENTER, para confirmá-lo. Pressionar a tecla ESC para finalizar a operação.</td></tr>
<tr><td>8
E_REG</td><td>Usada para verificar, ou alterar os registros específicos de R0 a R7 do banco de registros selecionado.

Exemplo: Para verificar o conteúdo do registro, pressionar a tecla E_REG. O display apresentará a solicitação do número do banco a ser alterado ou consultado.

B a n c o (0 - 3) _

Entrando com o número desejado, o display solicitará o número do registro que se deseja selecionar (neste exemplo, 1).

R (0 - 7) _

Entre com o número de registro desejado (no exemplo, 4). O conteúdo do registro solicitado será apresentado.</td></tr>
</table>

<table>
<tr><td></td><td>

```
R 4 = C 8
```

*Se for desejada apenas uma verificação, pressionar a tecla **ESC** para finalizar. Se for desejada uma alteração do conteúdo, entrar com o novo valor e pressionar a tecla **ENTER** para confirmá-lo.*
</td></tr>
<tr><td>A
REG_ESP</td><td>

Usada para verificar, ou alterar os registros especiais do 8031, acessados pelos endereços de 80h até FFh.

***Exemplo:** Pressionando-se a tecla **REG_ESP** será solicitado o endereço do registro especial.*

```
E n d :
```

*Entre com o endereço do registro e pressione a tecla **ENTER**. O endereço do registro e seu conteúdo são apresentados (no exemplo, 82).*

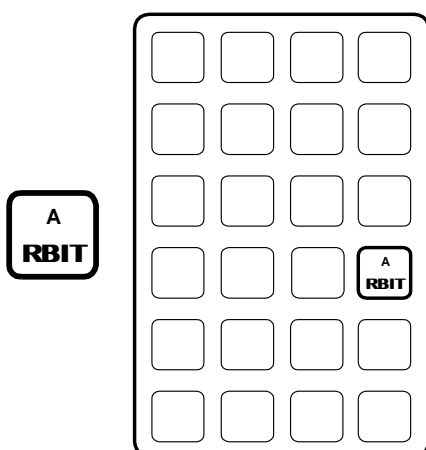
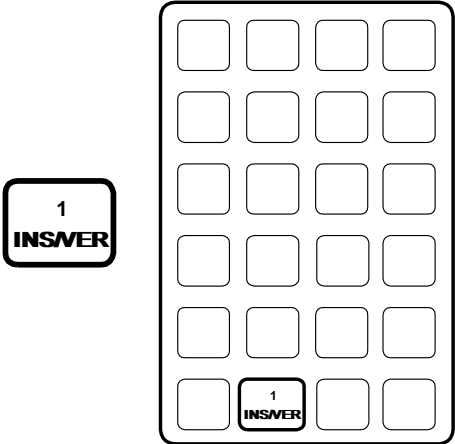
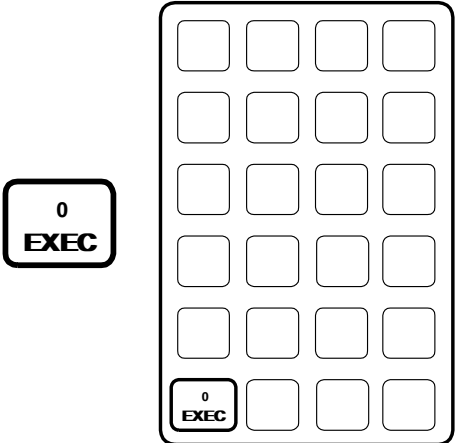
```
8 2   0 0
```

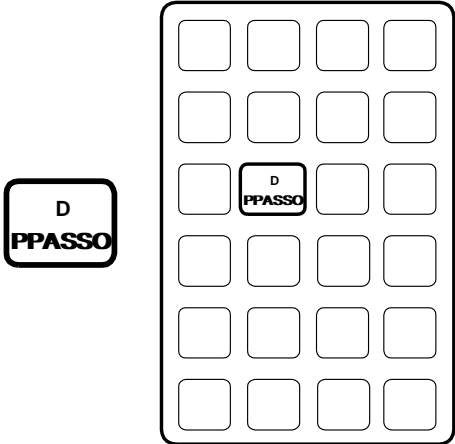
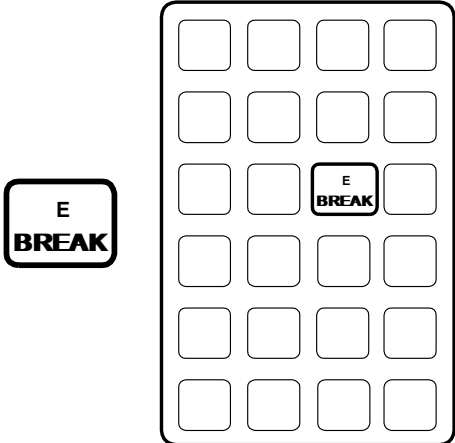
*Se for desejada apenas uma verificação do conteúdo, pressionar a tecla **ESC** para finalizar. Se for desejada uma alteração, entrar com o novo valor (no exemplo, 56).*

Com a entrada do novo valor, o endereço do registro será incrementado automaticamente e no display serão apresentados o próximo endereço e seu correspondente conteúdo.

```
8 2   5 6
8 3   0 0
```

*As teclas de funções auxiliares ↑ e ↓ serão usadas, respectivamente, para decrementar e incrementar os endereços dos registros, sem alterar os seus conteúdos. Para finalizar a verificação ou alteração dos registros especiais pressionar a tecla **ESC**.*
</td></tr>
</table>

Tecla	Descrição

A
RBIT
	Usada para verificar, ou alterar, os bits endereçáveis dos registros de funções especiais, ou os bits endereçáveis do bloco de memória interna.
	
	Os bits endereçáveis do bloco de memória interna são acessados pelos endereços de bit de 00h até 7Fh e ocupam os bytes de endereços 20h até 2Fh, num total de 128 bits endereçáveis.
	
	Os bits endereçáveis dos registros de funções especiais são acessados pelos endereços de bit de 80h até FFh e ocupam os bytes de endereçamento terminado por 0h, ou por 8h. Por exemplo, bytes 80h, 88h, 90h, A8h, etc. Assim os endereços de bit 80h até 87h acessam, respectivamente, os bits de 0 até 7 do endereço de byte 80h. Os endereços de bit 88h até 8Fh acessam respectivamente os bits de 0 até 7 do endereço de byte 88h. E assim sucessivamente.
	
	Exemplo: *Ao pressionar a tecla* ***RBIT*** *será solicitado o endereço do bit a ser acessado, de 00h até FFh.*
	
	`Bit(00-FF)_`
	
	Entrar com o endereço do bit desejado (no exemplo, 56) e pressionar ***ENTER****. Será apresentado o endereço do bit, seu valor, o endereço do byte que contém este bit e a posição do bit neste byte.*
	
	`56  0`
	`Bit  2A.6`
	
	No exemplo, o bit de endereço 56, cujo valor é zero, está localizado no byte de endereço 2A, da região de memória interna, e é o bit 6 deste endereço.
	
	Se for desejada apenas uma verificação, pressionar a tecla ***ESC*** *para finalizar. Se for desejada uma alteração, entrar com o valor do bit, 0 ou 1, e pressionar a tecla* ***ESC*** *para finalizar a função.*

2.5 FUNÇÕES DE COMANDO

<table>
<tr>
<td>3
ENCHER</td>
<td>Usada para o preenchimento de uma área de memória, desde o endereço inicial até o endereço final, com o valor um constante especificado.

Exemplo: Ao pressionar a tecla ENCHER, será solicitado a área de memória a ser preenchida.

0 - Ram Interna
1 - Ram Externa

Entrar com o valor 0 para selecionar a área de Ram Interna, endereços de 00h a 7Fh, ou com o valor 1 para selecionar a área de Ram Externa, endereços de 0000h a FFFFh.

Em ambos os casos, entrar com o endereço inicial e pressionar ENTER. Entrar com o endereço final e pressionar ENTER. Finalmente, entrar com o valor do byte, a ser utilizado no preenchimento, e pressionar ENTER.</td>
</tr>
<tr>
<td>2
MOV_BLOC</td>
<td>Usada para mover blocos de memória, de uma região especificada por um endereço inicial fonte e pelo número de bytes da região, para uma região de memória especificada por um endereço inicial de destino.

Exemplo: Ao pressionar a tecla MOV_BLC, será solicitado a área de memória a ser usada.

0 - Ram Interna
1 - Ram Externa

Selecionar se a região de memória a ser usada será a interna (0), ou a externa (1). Entrar com o endereço inicial da região fonte e pressionar ENTER. Entrar com o número de bytes a ser movimentado e pressionar ENTER. Finalmente, entrar com o endereço inicial do destino e pressionar ENTER.

Para abortar o processo durante a execução da função, basta pressionar a tecla ESC.</td>
</tr>
</table>

<table>
<tr>
<td>1
INS/VER</td>
<td>

Usada para inserir, ou verificar dados na Ram interna ou externa.

No sistema SDM 9431, a memória de programas e a memória de dados endereçam uma região comum de memória, a fim de possibilitar o desenvolvimento de programas. Assim, esta tecla também é usada para edição de programas.

***Exemplo:** Ao pressionar esta tecla, será solicitada a seleção da região de memória a ser acessada.*

```
0 - Ram Interna
1 - Ram Externa
```

*Após selecionar a região de memória, será solicitado o endereço inicial de acesso, de **00**h a **7F**h para a Ram interna e de **0000**h aa **FFFF**h para a Ram externa. Entrar com o endereço e pressionar a tecla **ENTER**. Assim será apresentado no display o endereço e o seu correspondente conteúdo.*

As teclas de funções auxiliares ↑ e ↓ serão usadas, respectivamente, para decrementar e incrementar os endereços acessados, sem a alteração dos seus conteúdos.

*Para alterar o conteúdo da memória selecionada, entrar com o novo valor e pressionar **ENTER**. Após isso, o endereço será automaticamente incrementado. Para finalizar ou abortar o comando pressionar a tecla **ESC**.*

</td>
</tr>
<tr>
<td>0
EXEC</td>
<td>

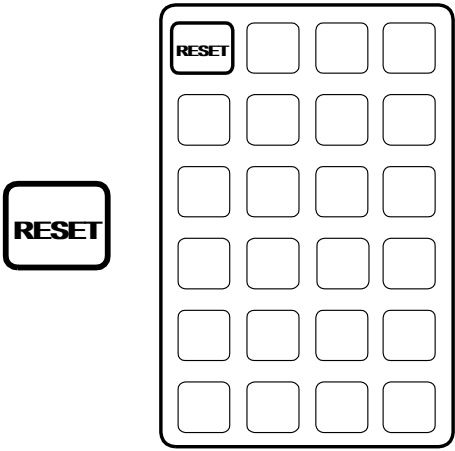
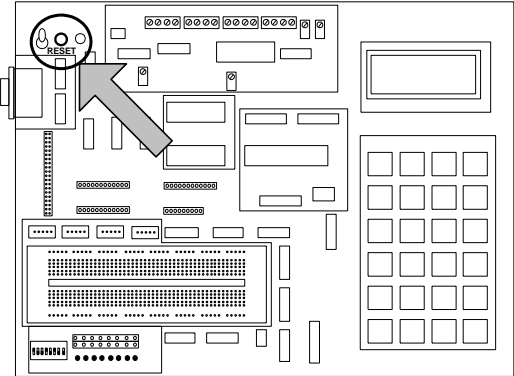
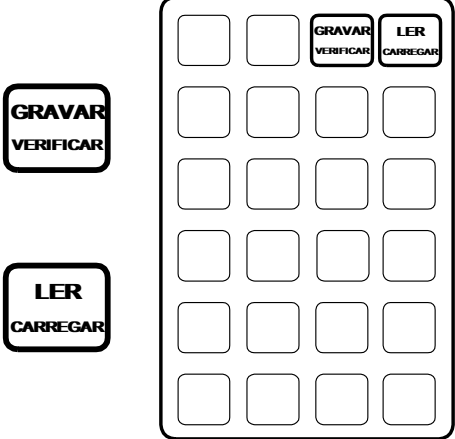
Usada para executar um programa em tempo real.

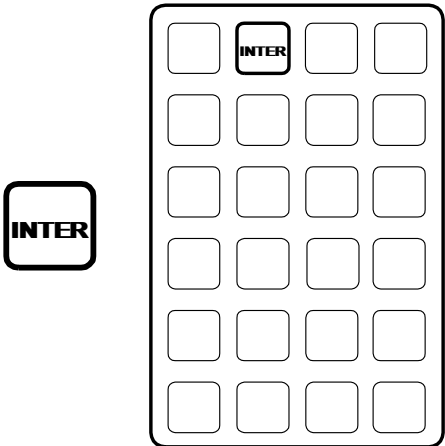
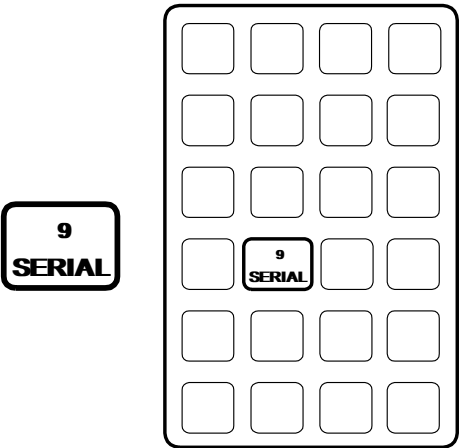
Programas em loop serão executados indefinidamente, até ocorrer uma interrupção. Programas com término definido deverão encerrar com um retorno ao programa monitor do sistema, ou seja, **LCALL 01C0**h.

***Exemplo:** Ao pressionar a tecla **EXEC** será solicitado o endereço inicial de execução.*

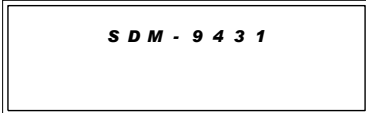
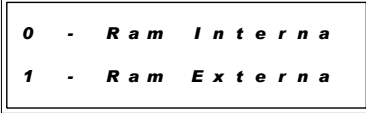
```
END exec: _
```

*Entrar com o endereço e pressionar a tecla **ENTER**.*

</td>
</tr>
</table>

D PPASSO	Usada para a execução de um programa no modo passo a passo, ou seja, executa somente a instrução do endereço apontado pelo contador de programa. **Antes de utilizar esta função, o contador de programa (PC) deve ser carregado com o endereço inicial de execução.** ***Exemplo:*** *Ao pressionar este tecla, no display serão mostrados os conteúdos do PC, do acumulador e do SP.* A C C = 0 0 S P = 0 7 P C = 0 0 0 0 _ *Pressionar a tecla **ENTER** para executar uma instrução. No display são atualizados os novos valores dos registros PC, ACC e SP. Pressionando-se sucessivamente a tecla **ENTER**, haverá sucessivas execuções de instruções.* *Após a execução de uma instrução, pode-se examinar ou alterar os conteúdos de memória ou registros. Para isto pressionar a tecla **ESC**, examinar o registro ou memória desejada e retornar à execução passo a passo, pressionando-se novamente a tecla **PPASSO**. A tecla **ESC** finaliza a operação.*
E BREAK	Usada para atribuir um ponto de parada (**BREAKPOINT**) na execução do programa, a fim de possibilitar depurações em velocidades maiores que a do modo passo a passo. **Antes de utilizar esta função, o contador de programa (PC) deve ser carregado com o endereço inicial de execução.** ***Exemplo:*** *Ao pressionar a tecla **BREAK** uma mensagem de solicitação de endereço de parada será apresentada.* E X E C A T E : _ *Entrar com o endereço e pressione a tecla **ENTER**. O endereço sempre deve corresponder a uma instrução, nunca a um dado complementar de uma instrução.*

NOTAS

1. A execução do programa no modo ***breakpoint*** utiliza a interrupção externa **INT1** para ser implementada. Assim, no modo ***breakpoint*** a interrupção **INT1** não poderá ser utilizada e o jump **JP5**, deverá estar posicionado para o lado esquerdo da conexão (entre os pinos identificados por GND e INT1).
2. Na utilização normal da interrupção externa INT1, o jump JP5 deverá estar posicionado para o lado direito da conexão (entre os pinos INT1 e INT1#).

2.6 FUNÇÕES DO SISTEMA

As teclas seguintes serão usadas para as operações de funções do sistema, que incluem Reset, interrupções, comunicação serial e gravador de EPROMs.

RESET	Usada para efetuar uma inicialização do sistema. Esta tecla está em paralelo com a botoeira ***RESET,*** localizada no canto superior esquerdo do módulo.
GRAVAR VERIFICAR LER CARREGAR	Usadas em conjunto com o gravador de EPROM do sistema SDM 9431, fornecido opcionalmente. A operação das mesmas é descrita no manual do SDM-EPROM, gravador de EPROMs do sistema SDM 9431.

Tecla	Descrição
INTER	Usada para exemplificar o uso de interrupção no sistema SDM-9431. Esta tecla está conectada à entrada de interrupção **INT0** do microcontrolador 8031. Para a sua utilização deve ser habilitado o uso da interrupção 0 e ser especificado o endereço vetorial da rotina de serviço de interrupção 0. ***Exemplo:*** *No item 6.17 deste manual é apresentado um programa exemplo que utiliza esta função.*
9 SERIAL	Usada para carregar, ou salvar, uma região de memória externa em uma unidade de disquete de um computador compatível com IBM-PC, conectado ao sistema SDM 9431 através da porta serial. Este comando somente poderá ser utilizado se existir um cabo de conexão entre o módulo SDM 9431 e a porta serial do computador PC e utilizando o software fornecido pela DATAPOOL, descrito na seção 05 deste manual.

2.7 EXEMPLO DE UTILIZAÇÃO DO MÓDULO SDM-9431 NO MODO TECLADO

A tabela seguinte apresenta a seqüência de operações para inserir e executar um programa no modo teclado.

TECLA DIGITADA	DISPLAY	COMENTÁRIO
RESET	S D M - 9 4 3 1	Depois da execução de qualquer comando ou de um RESET, o sistema mostra a mensagem e aguarda novo comando.
1 INS/VER	0 - Ram Interna 1 - Ram Externa	O usuário deve escolher em qual tipo de Ram deseja inserir dados.

Tecla	Display	Descrição
1 INS/VER	End : _	Após optar pela Ram externa, o usuário terá que dar o endereço inicial para inserir seu programa.
5 DPTR, 0 EXEC, 0 EXEC, 0 EXEC	End : 5000_	O endereço deverá ser de **5000**h à **BFFF**h ou de **5000**h à **5FFF**h, dependendo da RAM utilizada.
ENTER	5000 FF	Após apertar ENTER, o sistema fica esperando que o usuário entre com o seu programa.
7 PSW, 5 DPTR	5000 75 5001 FF	Inserindo o programa.
8 E_REG, 1 INS/VER	5001 81 5002 FB	
2 MOV_BLOC, F	5002 2F 5003 FF	
1 INS/VER, 2 MOV_BLOC	5003 12 5004 FF	
1 INS/VER, 0 EXEC	5004 10 5005 EE	Inserindo o programa.
A REG_ESP, A REG_ESP	5005 AA 5006 F5	
1 INS/VER, 2 MOV_BLOC	5006 12 5007 FF	
1 INS/VER, 0 EXEC	5007 10 5008 EE	
0 EXEC, 2 MOV_BLOC	5008 02 5009 F5	

Teclas	Display	
3 ENCHER, 3 ENCHER	5009 33 500A F5	
4 ACC, 0 EXEC	500A 40 500B FF	
0 EXEC, 9 SERIAL	500B 09 500C 9F	
1 INS/VER, 3 ENCHER	500C 13 500D FE	
1 INS/VER, 2 MOV_BLOC	500D 12 500E FD	
1 INS/VER, 0 EXEC	500E 10 500F FF	
A REG_ESP, A REG_ESP	500F AA 5010 EE	
1 INS/VER, 2 MOV_BLOC	5010 12 5011 F9	Inserindo o programa.
1 INS/VER, 0 EXEC	5011 10 5012 FD	
E BREAK, 7 PSW	5012 E7 5013 EF	
8 E_REG, 0 EXEC	5013 80 5014 FF	
F, 1 INS/VER	5014 F1 5015 FB	
1 INS/VER, 2 MOV_BLOC	5015 12 5016 E6	

Tecla	Display	Descrição
0 EXEC, 1 INS/VER	5016 01 5017 8F	
C PC, 0 EXEC	5017 C0 5018 9E	
ESC -	SDM-9431	O sistema fica esperando um novo comando.
0 EXEC	END exec: _	O usuário deve entrar com o endereço inicial do seu programa.
5 DPTR, 0 EXEC, 0 EXEC, 0 EXEC	END exec: 5000_	Digitando endereço.
ENTER	_	Executa o programa do usuário e retorna para o programa monitor.
9 SERIAL C PC ENTER	09 0C SDM-9431	Se a tecla pressionada for de 0 até F, o valor será apresentado no display. Se for qualquer outra tecla, o programa será abortado, retornando ao programa monitor.

A seguir é apresentada a listagem mnemônica do programa:

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, #2Fh	inicializa o apontador de pilha (stack pointer)
5003	12 10 AA		lcall clr_dsp	limpa display
5006	12 10 02	início:	lcall le_tec	espera tecla pressionada
5009	33		rlc a	desloca Acc para a direita

500A	40 09		jc FIM	se sim final de programa
500C	13		rrc a	desloca Acc à esquerda
500D	12 10 AA		lcall clr_dsp	limpa display
5010	12 10 E7		lcall ac_dsp	Acc → display
5013	80 F1		sjmp início	retorna ao inicio
5015	12 01 C0	FIM:	lcall monitor	retorna para o programa monitor

CAPÍTULO 3 - PERIFÉRICOS EXTERNOS DO SISTEMA SDM 9431

Além das portas paralelas, porta serial e temporizadores, periféricos internos da família 8051, no sistema SDM 9431 foram incorporados periféricos externos ao micro-controlador. São eles: o teclado, o display de cristal líquido, um conversor D/A de um canal de oito bits e um conversor A/D de oito canais multiplexados de oito bits.

3.1 TECLADO

O teclado do sistema SDM 9431 é composto por 24 teclas agrupadas em uma matriz de 8 x 3, conforme o esquema apresentado na figura 3.1.

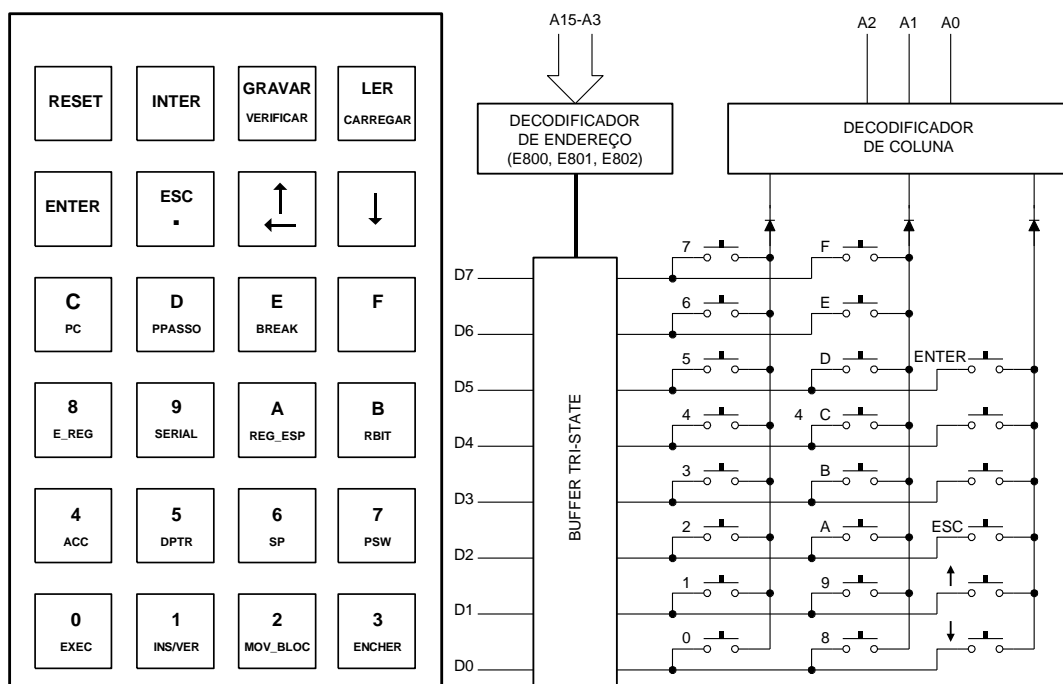



Figura 3.1 - Distribuição do teclado do SDM 9431.

O teclado é varrido através de leituras dos endereços E800h, E801h e E802h. O valor lido em cada caso identificará a tecla pressionada. A decodificação do teclado foi implementada parcialmente, conforme a figura 3.2 e, portanto, outros endereços na faixa de E800h até EBFFh também acessam o teclado.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	1	0	1	0	–	–	–	–	–	–	–	X	X	X

Figura 3.2 - Decodificação parcial para o teclado.

OBSERVAÇÕES:

1. A sub-rotina ***LE_TEC*** é usada para identificar a tecla pressionada e colocar no acumulador o valor correspondente a esta tecla.
2. As teclas **INTR** e **RESET** não fazem parte da matriz do teclado.
3. Também as sub-rotinas **LE_DADO** e **LE_DAD1** estão disponíveis para acesso ao teclado (vide Capítulo 4 – Sub-rotinas do Sistema SDM 9431).

3.2 DISPLAY DE CRISTAL LÍQUIDO

O display do sistema SDM 9431 é do tipo alfanumérico, composto de duas linhas de dezesseis colunas. O mesmo é de fácil interfaceamento e de consumo extremamente reduzido. A conexão do display ao microcontrolador foi realizado conforme o diagrama da figura 3.3.

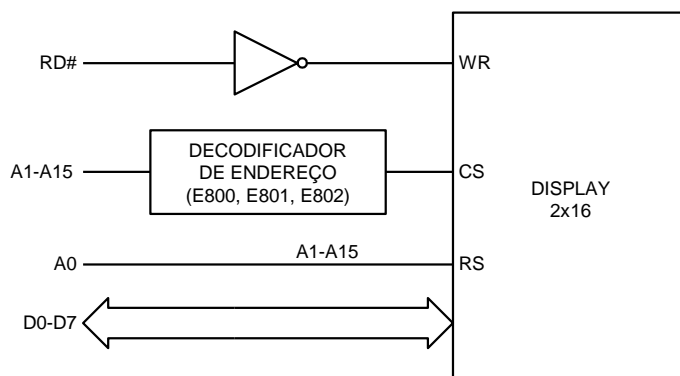



Figura 3.3 - Conexão entre display e microcontrolador.

A inicialização do modo de operação do display é realizada pelo programa monitor e acessada pela sub-rotina ***INI_DIS***, com endereço inicial em **1063**h. Esta inicialização define o modo de interfaceamento do display, o número de linhas do mesmo e o tipo da matriz do caracter. Após inicializado, o acesso ao display poderá ser realizado de duas maneiras: através de códigos de comando, ou através de escrita de dados. Os códigos de comando são enviados para o endereço **EC00**h e realizados pela sub-rotina ***DSP_COM***. Estes códigos correspondem aos comandos de operações no display, ou aos endereços de caracteres do se deseja acessar.

A decodificação do display foi implementada parcialmente, conforme a figura 3.4 e, portanto, outros endereços na faixa de **EC00**h até **EFFF**h também acessam o display.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	1	0	1	1	–	–	–	–	–	–	–	–	–	X

Figura 3.4 - Decodificação parcial para o display.

CARACTER

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
LINHA 1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
LINHA 2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

Figura 3.5 - Endereçamento dos caracteres do display.

A tabela 3.1 apresenta os comandos disponíveis para operações no display. Para utilização de um comando, o código do mesmo deverá ser carregado no acumulador e chamada a sub-rotina ***DSP_COM***.

DESCRIÇÃO DO COMANDO	MODO	CÓDIGO HEXADECIMAL
Controle do display	Ativo (lig.) s/ cursor	0C
	Desligado	0A, 08
Limpeza do display com retorno do cursor		01
Retorno do cursor à primeira posição da primeira linha e da mensagem à sua posição original		02
Controle do Cursor	Ativo (ligado-fixo)	0E
	Inativo	0C
	Alternado	0F
	Desl. à esquerda	10
	Desl. à direita	14
	Retorno	02
	Piscante	0D
Sentido de deslocamento do cursor na entrada de um novo caracter	À esquerda	04
	À direita	06
Deslocamento da mensagem na entrada de um novo caracter	À esquerda	07
	À direita	05
Deslocamento da mensagem sem a entrada de novos caracteres	À esquerda	18
	À direita	1C
Endereços das primeiras posições	Primeira linha	80
	Segunda linha	C0

Tabela 3.1 - Comandos para o display.

		4 BITS SUPERIORES											
		0	2	3	4	5	6	7	A	B	C	D	E
4 BITS INFERIORES		0000	0010	0011	0100	0101	0110	0111	1011	1100	1101	1110	1111
0	xxxx0000		0	@	P	`	p		ー	タ	ミ	α	p
1	xxxx0001	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
2	xxxx0010	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
3	xxxx0011	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
4	xxxx0100	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
5	xxxx0101	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
6	xxxx0110	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
7	xxxx0111	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
8	xxxx1000	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
9	xxxx1001	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
A	xxxx1010	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
B	xxxx1011	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
C	xxxx1100	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
D	xxxx1101	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
E	xxxx1110	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
F	xxxx1111	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

Tabela 3.2 - Código ASCII para o display.

Para acessar um caracter do display é necessário enviar o código de seu endereço para a sub-rotina ***DSP_COM***, ou seja, carregar o acumulador com o código do endereço e chamar a sub-rotina ***DSP_COM***. Logo após, o dado enviado para o display será apresentado na posição selecionada.

Para apresentar um dado no display, o seu valor ASCII deverá ser carregado no acumulador e chamar a sub-rotina ***DSP_DAT***. Os dados são escritos no display através do endereço **EC01**h. Várias sub-rotinas de acesso ao display foram desenvolvidas e estão descritas no Capítulo 4 – Sub-rotinas do Sistema SDM 9431.

3.3 O CONVERSOR DIGITAL-ANALÓGICO

O sistema SDM 9431 utiliza um DAC0800, conversor digital-analógico de oito bits com tempo de conversão de 100ns. O diagrama de blocos do circuito de interface do mesmo é apresentado na figura 3.6.

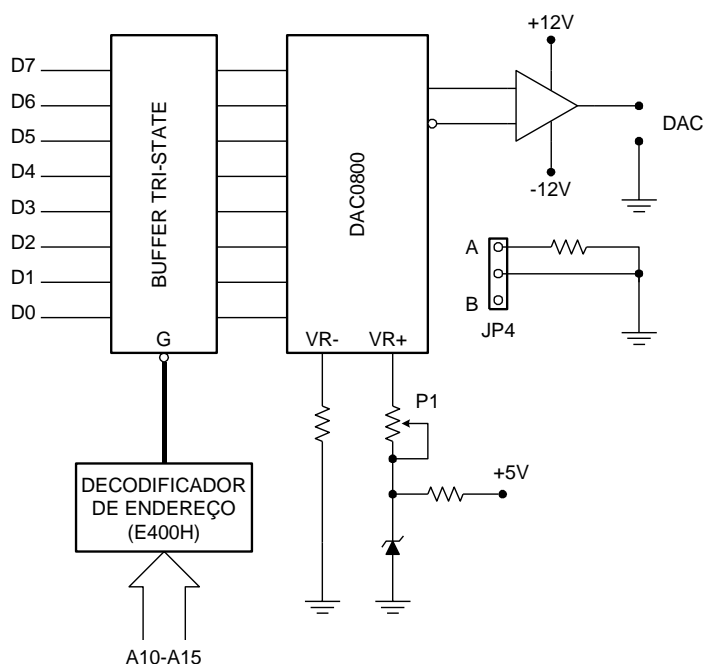



Figura 3.6 - Interface com o DAC 0800.

O conversor será acessado por uma escrita no endereço E400h. Esta decodificação é parcial, conforme apresentado na figura 7 e, portanto, outros endereços na faixa de E400h a E7FFh também acessam o conversor DA.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	1	0	0	1	–	–	–	–	–	–	–	–	–	–

Figura 3.7 - Decodificação parcial para o conversor digital analógico.

NOTA

1. O trimpot P1 é usado para ajustar o conversor.
2. O jump JP4 é usado para efetuar um conversão de 0V a +12V, quando posicionado para o lado A, ou de -12V a +12V, quando posicionado para o lado B.
3. A subrotina ***DA*** implementa o controle do DAC (vide Capítulo 4 – Sub-rotinas do Sistema SDM 9431).

3.4 O CONVERSOR ANÁLOGO-DIGITAL

O sistema SDM 9431 utiliza um ADC0808, conversor análogo-digital oito bits, que contém oito canais multiplexados e usa a técnica de conversão por aproximação sucessiva. O tempo de conversão de 100 µs permite uma alta velocidade de conversão, com alta precisão e com uma mínima dissipação de potência. O diagrama de blocos do circuito de interface do mesmo está apresentado na figura 3.8.

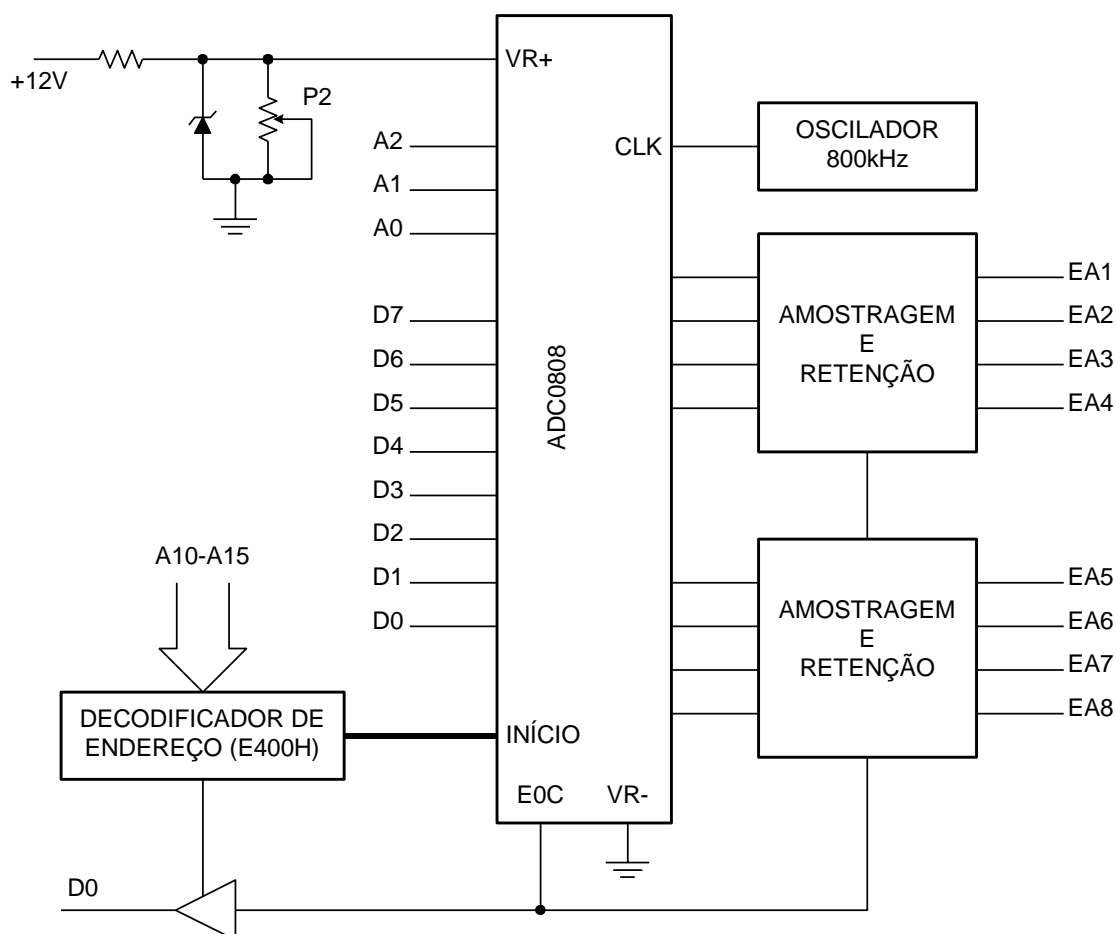



Figura 3.8 - Interface com o ADC 0808.

Os oito canais do conversor serão acessados pelos endereços de E000h a E007h. Esta decodificação é parcial, conforme apresentado na figura 3.9 e, portanto, outros endereços na faixa de E000h a E3FFh também acessam o conversor AD.

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1	1	1	1	1	1	–	–	–	–	–	–	–	x	x	x

Figura 3.9 - Decodificação parcial para o conversor digital analógico.

NOTA

1. O início da conversão é realizado pela escrita no endereço de qualquer um dos canais do conversor. O processo de conversão será interrompido se ocorrer um novo início de conversão.
2. A monitoração do sinal EOC, fim de conversão, realizada através da leitura do endereço E400h, possibilita a realização de experiências com conversões contínuas.
3. Um oscilador, com frequência entre 800kHz e 1MHz, é usado para temporização da conversão.
4. O potenciômetro P2 permite o ajusto do conversor.
5. A subrotina ***AD*** implementa o controle do ADC (vide Capítulo 4 – Sub-rotinas do Sistema SDM 9431).

CAPÍTULO 4 - SUB-ROTINAS DO SISTEMA SDM 9431

4.1 SUB-ROTINAS PARA O TECLADO

As sub-rotinas disponíveis que acessam ao teclado são:

4.1.1 LE_TEC

LE_TEC	
DESCRIÇÃO	Espera uma tecla ser pressionada e retorna o valor da tecla no acumulador.
CHAMADA	LCALL 1002h
PARÂMETROS DE SAÍDA	Tecla pressionada → ACC
PARÂMETROS DE ENTRADA	nenhum
REGISTROS ALTERADOS	ACC
COMENTÁRIOS	O valor lido será de 00h até 0Fh para as teclas de dados e de F0h até F5h para as teclas de comando, ou teclas auxiliares.

4.1.2 LE_DADO

LE_DADO	
DESCRIÇÃO	Espera por duas teclas de dados (0 a F) pressionadas, sem tecla de confirmação (veja **LE_DAD1**). As duas teclas pressionadas formarão o byte que será transferido para o acumulador.
CHAMADA	LCALL 0F00h
PARÂMETROS DE SAÍDA	1 Byte = 2 teclas → ACC
PARÂMETROS DE ENTRADA	nenhum
REGISTROS ALTERADOS	ACC
COMENTÁRIOS	Se a tecla ESC for pressionada o carry será setado (C = 1). Isto poderá ser usado como condição de teste para fim de entrada de dados pelo teclado.

4.1.3 LE_DAD1

LE_DAD1	
DESCRIÇÃO	Espera por duas teclas de dados (0 a F) pressionadas, com tecla de confirmação (veja **LE_DADO**). As duas teclas pressionadas formarão o byte que será transferido para o acumulador.
CHAMADA	LCALL 0F27h
PARÂMETROS DE SAÍDA	1 Byte = 2 teclas $\rightarrow$ ACC
PARÂMETROS DE ENTRADA	nenhum
REGISTROS ALTERADOS	ACC
COMENTÁRIOS	Sua operação é idêntica à descrita em ***LE_DADO***. Difere somente na necessidade de confirmação das teclas pressionadas, ou seja, após duas teclas de dados (0 a F) pressionadas há a necessidade de pressionar a tecla ENTER para o retorno da sub-rotina.

4.2 SUB-ROTINAS PARA O DISPLAY

4.2.1 CLR_DSP

CLR_DSP	
DESCRIÇÃO	A sub-rotina ***CLR_DSP*** limpa o display e coloca o cursor na primeira posição da primeira linha.
CHAMADA	LCALL 10AAh
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	nenhum
REGISTROS ALTERADOS	nenhum

4.2.2 AC_DSP

AC_DSP	
DESCRIÇÃO	***AC_DSP***
CHAMADA	LCALL 10E7h
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	ACC
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	A sub-rotina ***AC_DSP*** apresenta o conteúdo do acumulador no display **Exemplo:** Se mov a, # 35h Lcall 10E7h então, o display mostrará "35".

4.2.3 DPT_DSP

DTP_DSP	
DESCRIÇÃO	apresenta o conteúdo do registro DPTR no display
CHAMADA	LCALL 1121h
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	DPTR
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	**Exemplo:** Se mov dptr, # 5272h LCALL 1121h então, o display mostrará "5272".

4.2.4 MENS

MENS	
DESCRIÇÃO	Apresenta uma mensagem no display
CHAMADA	LCALL 110Fh
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	DPTR deve conter endereço da mensagem
REGISTROS ALTERADOS	DPTR
COMENTÁRIOS	O registro DPTR deverá apontar para o endereço inicial da mensagem. O primeiro byte da mensagem deverá conter o número de caracteres da mesma, seguido dos correspondentes códigos ASCII dos caracteres. **Exemplo:** MEM1: db 14, 'Teste_de_saída' onde o número 14 corresponde ao número de caracteres da mensagem. Esta mensagem residirá na memória com os seguintes códigos: OE_54_65_73_74_65_20_64_65_20_73_61_69_64_61 14 T e s t e d e S a í d a Assim, a chamada será efetuada por: mov dptr, # MEM1 (endereço de MEM1) LCALL 110Fh.

4.2.5 DSP-COM

DSP_COM	
DESCRIÇÃO	envia um código de comando, colocado no acumulador, para o display. (vide Capítulo 3 - Periféricos Externos ao 8031)
CHAMADA	LCALL 109Ah
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	ACC
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	**Exemplo:** mov a, # 01h LCALL 109Ah irá limpar o display.

4.2.6 DSP-DAT

DSP_DAT	
DESCRIÇÃO	Envia um dado (ASCII), colocado no acumulador para o display.
CHAMADA	LCALL : 10FFh
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	ACC
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	A escrita do dado automaticamente incrementa a posição do cursor, ou seja, uma nova escrita será deslocada em relação a escrita anterior. **Exemplo:** mov a, # 66 LCALL 10FFh irá escrever a letra "f" na posição do cursor e incrementar a posição do cursor.

SUBROTINAS DE USO GERAL

4.3.1 ASCII

ASCII	
DESCRIÇÃO	Usada para receber um byte de acumulador e retornar com os correspondentes códigos ASCII deste byte nos registros R1 e R2.
CHAMADA	LCALL 114Ch
PARÂMETROS DE SAÍDA	R1, R2
PARÂMETROS DE ENTRADA	ACC
REGISTROS ALTERADOS	flag C, R1 e R2
COMENTÁRIOS	**Exemplo:** Se ACC = 65 então: LCALL 114Ch terá como resultado R1= 36 e R2 = 35.

4.3.2 AD

AD	
DESCRIÇÃO	Lê um sinal do conversor análogo digital, endereçado pelo registro DPTR, e carrega o valor lido no acumulador.
CHAMADA	LCALL 145Fh
PARÂMETROS DE SAÍDA	ACC
PARÂMETROS DE ENTRADA	DPTR
REGISTROS ALTERADOS	ACC
COMENTÁRIOS	O endereçamento dos canais EA1 até EA8 será feito respectivamente pelos endereços E000h até E007h.

4.3.3 DA

DA	
DESCRIÇÃO	Envia um dado, armazenado no acumulador, para o conversor digital analógico.
CHAMADA	LCALL 1471h
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	ACC
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	

4.3.4 DELAY

DELAY	
DESCRIÇÃO	Sub-rotina de atraso rápido, utilizada pelo programa monitor e que pode ser acessada pelo programa do usuário.
CHAMADA	LCALL 11C8h
PARÂMETROS DE SAÍDA	nenhum
PARÂMETROS DE ENTRADA	nenhum
REGISTROS ALTERADOS	nenhum
COMENTÁRIOS	Atrasos diferentes do apresentado por esta sub-rotina deverão ser criados pelo próprio usuário.

CAPÍTULO 5 - CONEXÃO COM MICROCOMPUTADORES PC

5.1 INTRODUÇÃO

O módulo SDM-9431 possui uma porta serial RS-232, que permite a comunicação do módulo com um microcomputador compatível com IBM-PC, disponível nas versões Windows e DOS.

A comunicação é feita pelo programa SDM-9431, elaborado pela Datapool e fornecido em conjunto o módulo didático, permite:

- salvar no microcomputador programas do módulo SDM-9431;
- carregar programas, armazenados no microcomputador, na memória do módulo SDM-9431;
- inspecionar e alterar registros, temporizadores e contadores do SDM-9431;
- inspecionar, alterar e preencher áreas de memória do SDM-9431;
- executar, em diversos modos, e acompanhar a execução de programas no SDM-9431.

5.2 INSTALAÇÃO DO PROGRAMA

Para instalar o programa SDM-9431 basta executar o programa ***Instalar***. O programa de instalação exibirá a janela mostrada na figura 5.1.

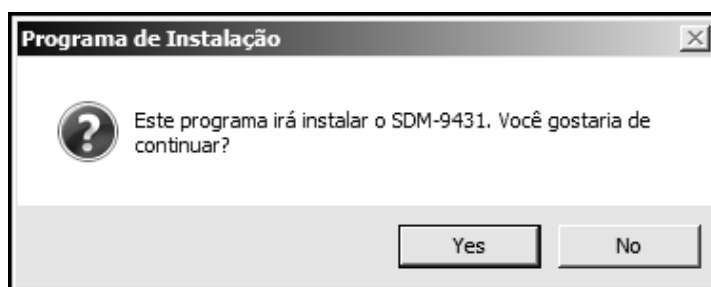



Figura 5.1 – Janela de abertura do programa de instalação do SDM-9431.

NOTA

A instalação do programa em microcomputadores com o sistema operacional Windows Vista poderá provocar o aparecimento de uma mensagem solicitando autorização do administrador do sistema para prosseguir a instalação.

Pressionando-se Sim (ou Yes), o programa de instalação exibirá a janela mostrada na figura 5.2.

Figura 5.2 – Tela de boas vindas do programa de instalação do SDM-9431.

Pressionando-se o botão ***Avançar >*** será exibida a janela mostrada na figura 5.3, onde o usuário poderá optar por instalar o programa SDM-9431 na pasta sugerida ou em outra de sua escolha.

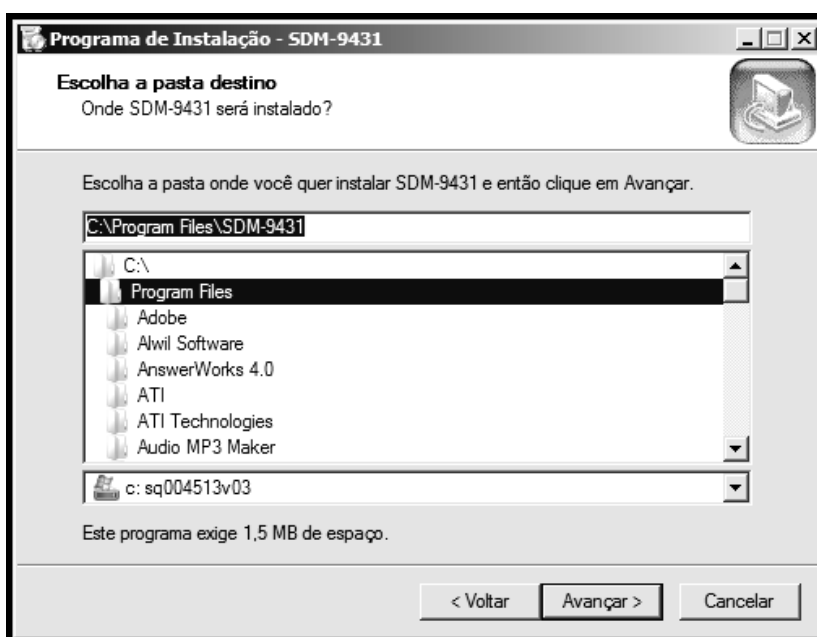



Figura 5.3 – Tela de escolha da pasta de instalação do SDM-9431.

Pressionando-se o botão ***Avançar >*** , o usuário poderá optar por criar a pasta SDM-9431 ou escolher uma pasta no Menu Iniciar do Windows para a instalação.

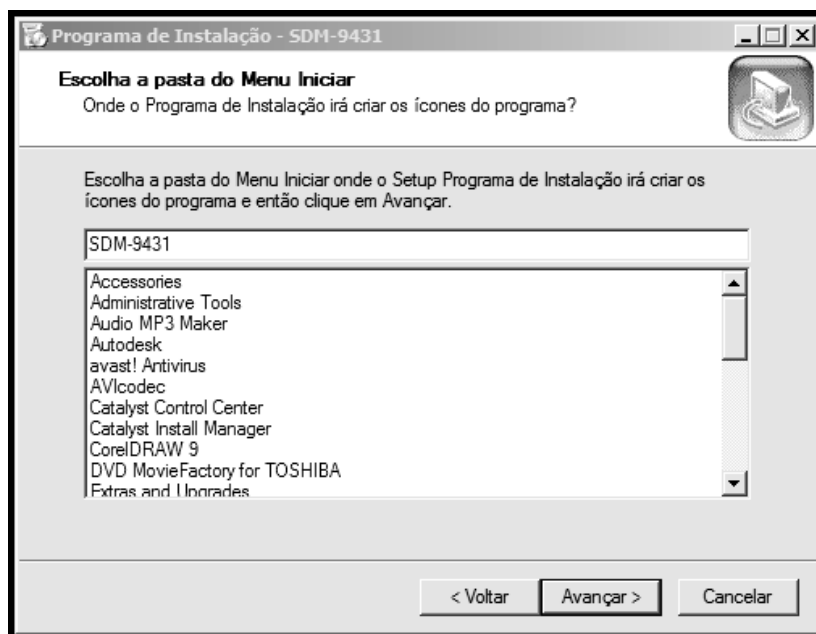



Figura 5.4 – Janela de escolha de pasta no Menu Iniciar.

Pressionando-se o botão ***Avançar >*** , o usuário poderá optar por criar um ícone de atalho para o programa na área de trabalho do Windows.

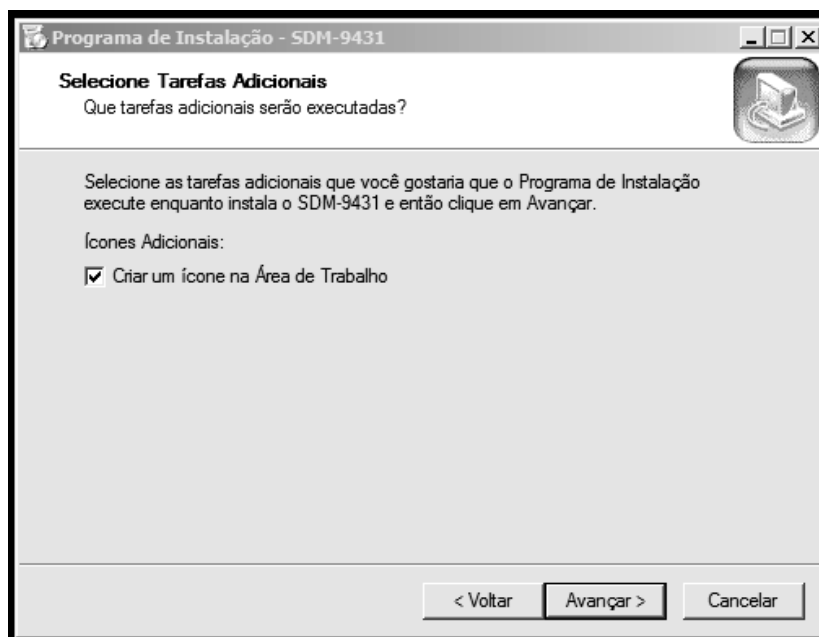



Figura 5.5 – Janela de opção da criação do ícone na área de trabalho.

Pressionando-se o botão ***Avançar >*** , o programa de instalação está pronto para iniciar o processo de instalação, exibindo a janela mostrada na figura 5.6.

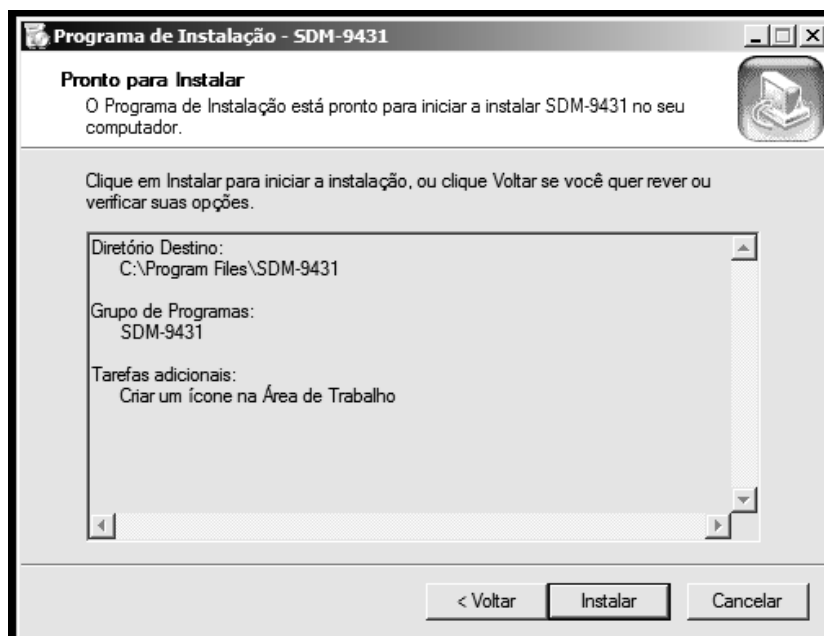



Figura 5.6 – Janela de início do processo de instalação do programa SDM-9431.

Pressionando-se o botão ***Instalar >*** , o programa de instalação será executado, exibindo a janela da figura 5.7, que mostra o progresso do processo de instalação.

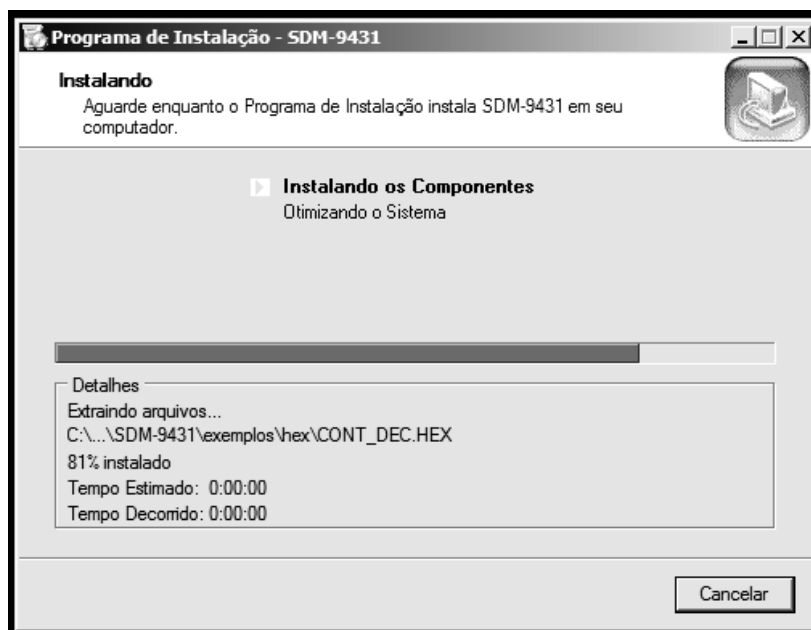



Figura 5.7 – Janela do processo de instalação do programa SDM-9431.

Após o término do processo, o programa de instalação exibe a janela da figura 5.8, onde o usuário poderá optar por executar o programa SDM-9431 imediatamente à finalização do programa de instalação. Pressionado o botão ***Concluir*** o programa de instalação será finalizado e, caso selecionado, iniciado o programa SDM-9431.

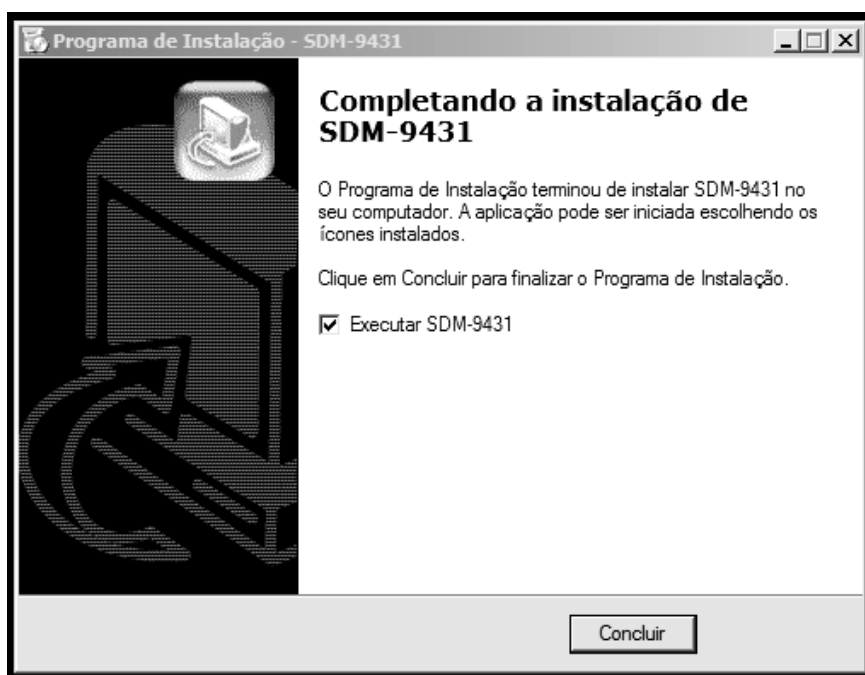



Figura 5.8 – Janela de finalização do programa de instalação.

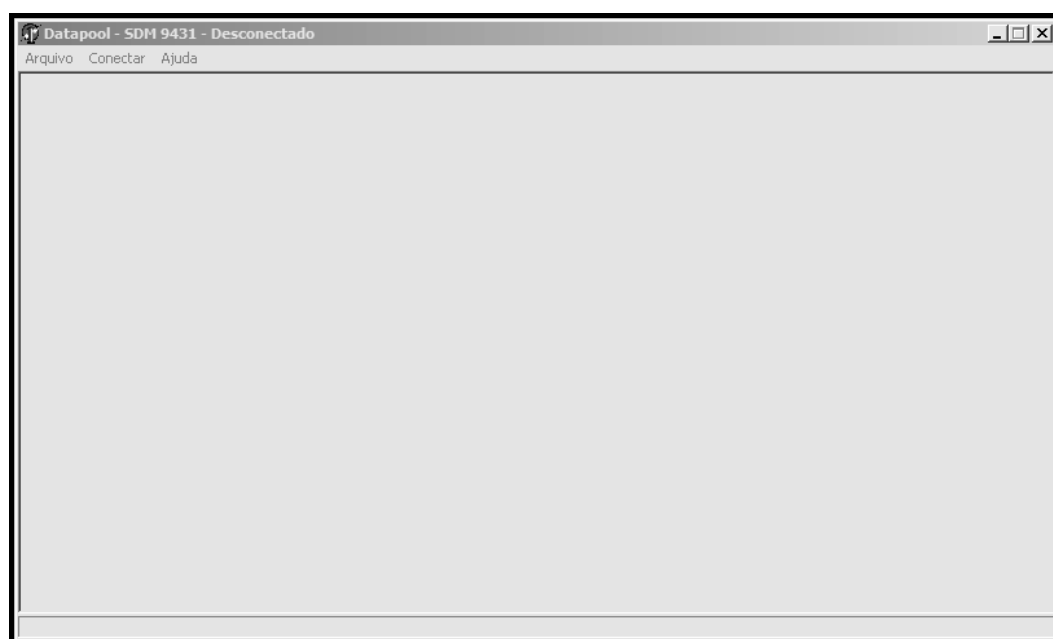



Figura 5.8 – Janela inicial do programa SDM-9431.

5.3 CONECTANDO O MÓDULO SDM-9431 AO PC

A conexão do módulo SDM-9431 ao PC é obtida através das seguintes etapas:

1. Conectar o módulo SDM-9431 ao PC usando um cabo serial, padrão RS-232.
2. Executar o programa SDM-9431. Será aberta a janela mostrada na figura 5.9. A palavra ***Desconectado*** na barra de título do programa, mostra que o módulo SDM-9431 ainda não está conectado ao PC.

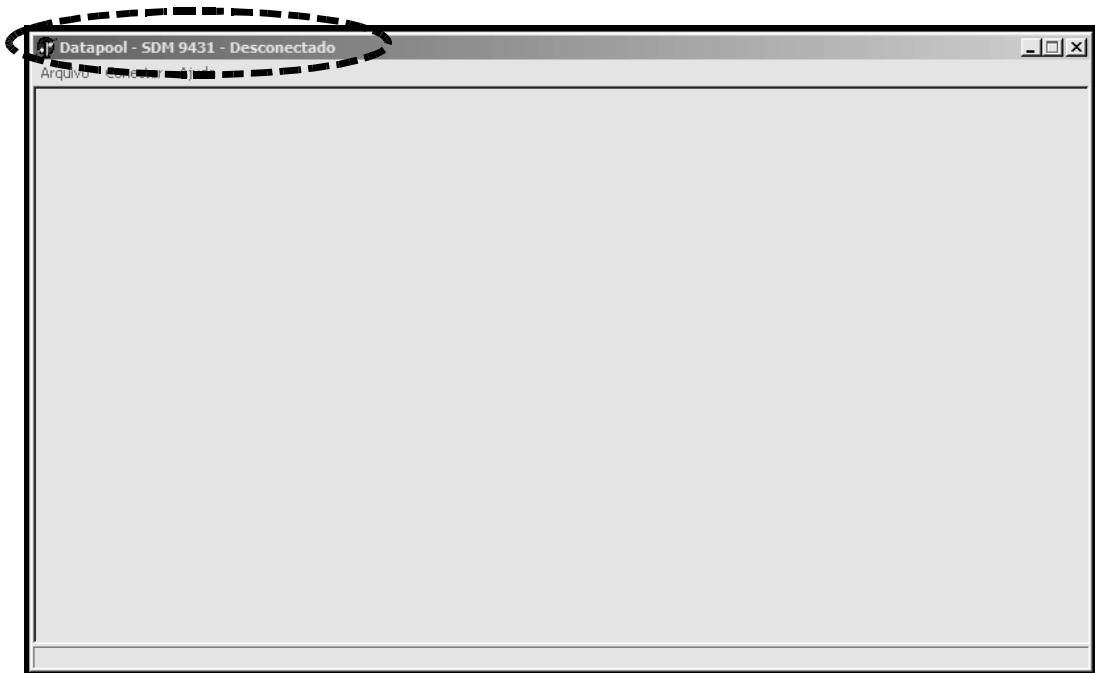



Figura 5.9 – Janela inicial do programa SDM-9431, em destaque a indicação do módulo desconectado.

3. A seguir, deve ser configurada a porta serial que o programa usará para conectar o módulo ao PC. Para tanto, selecionar a opção ***Configurar Serial ...*** no menu ***Conectar*** , conforme mostrado na figura 5.10. Selecionando esta opção será aberta uma janela, solicitando a escolha da porta onde estará conectado o módulo SDM-9431, conforme mostra a figura 5.11.

<u>NOTAS</u>

a) A cada vez que é feita a conexão deve-se selecionar a porta serial que será utilizada.

b) A identificação da porta serial a ser usada pode ser feita pelo Gerenciador de Dispositivos do Windows ou, simplesmente, por tentativas.

c) Caso haja dificuldades em se obter a conexão, procure ajuda com o suporte técnico de seu PC.

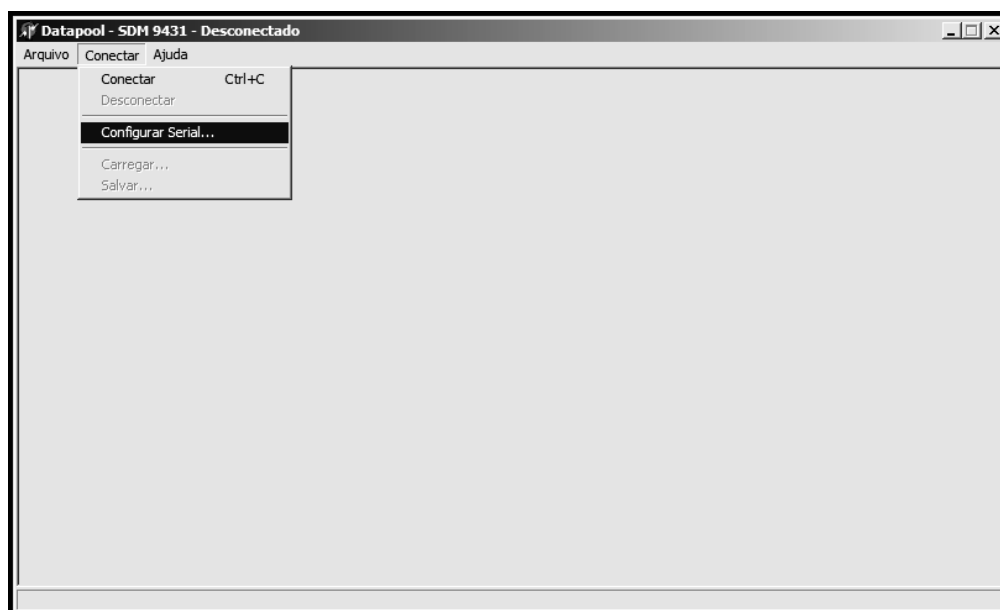



Figura 5.10 – Menu de seleção da porta seria.

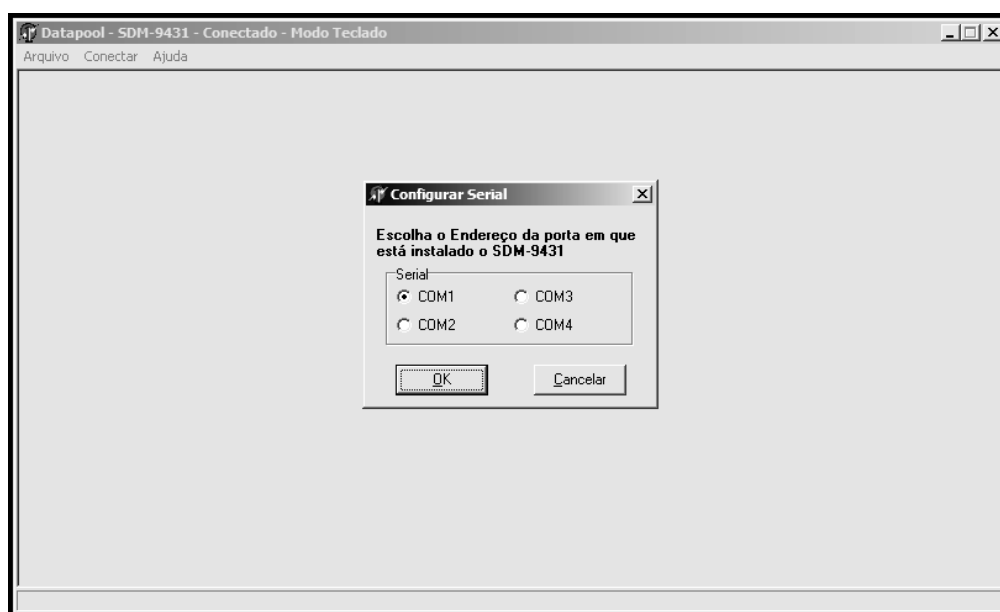



Figura 5.11 – Janela de seleção da porta serial.

4. Deve-se selecionar o modo de comunicação entre o módulo e o PC, usando a chave ***MODO*** do módulo SDM-9431, cuja localização é mostrada na figura 5.12.

   No modo ***TECLADO***, o PC servirá, basicamente, para salvar programas do módulo no PC e carregar programas do PC no módulo.

No modo ***PC***, o programa SDM-9431 dá acesso e permite alterar o conteúdo da memória do módulo e dos registros do microcontrolador, além de permitir o acompanhamento da execução de programas.

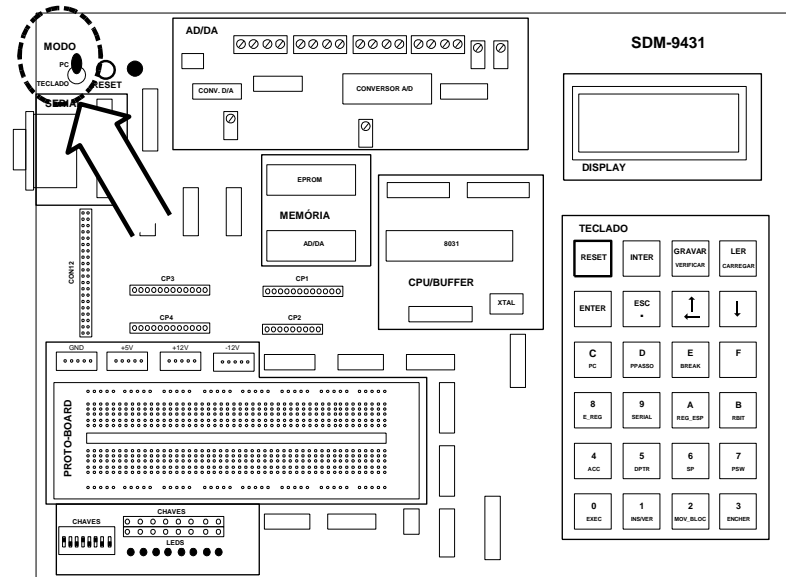



Figura 5.12 – Localização da chave de seleção de MODO (PC ou teclado).

5. O próximo passo é conseguir a comunicação do módulo SDM-9431 com o PC. Para tanto, deve-se selecionar a opção Conectar do menu Conectar, obtendo o aparecimento da mensagem mostrada na figura 5.13.

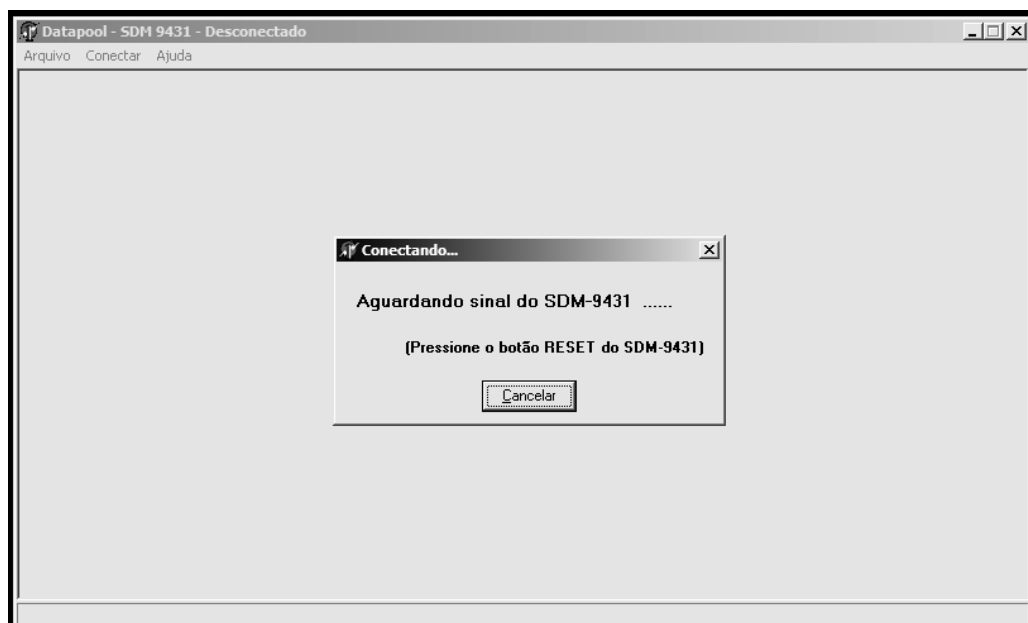



Figura 5.13 – Janela indicando que o programa está aguardando sinal do módulo e solicitando que seja pressionado um dos botões de RESET do módulo.

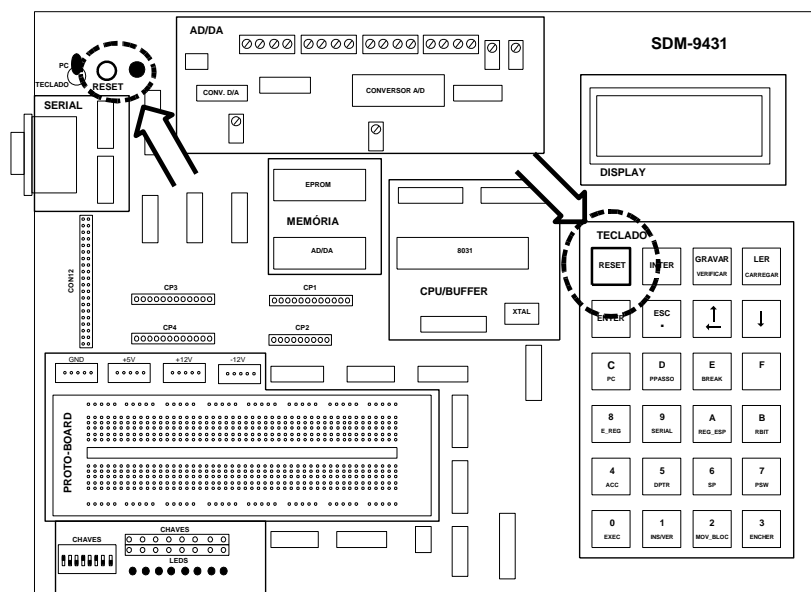



Figura 5.14 – Localização dos botões de RESET no módulo SDM-9431.

6. Pressionando-se um dos botões de ***RESET*** do módulo, a janela do programa assumirá o aspecto (com a chave ***MODO*** em ***PC***) mostrado na figura 5.15.

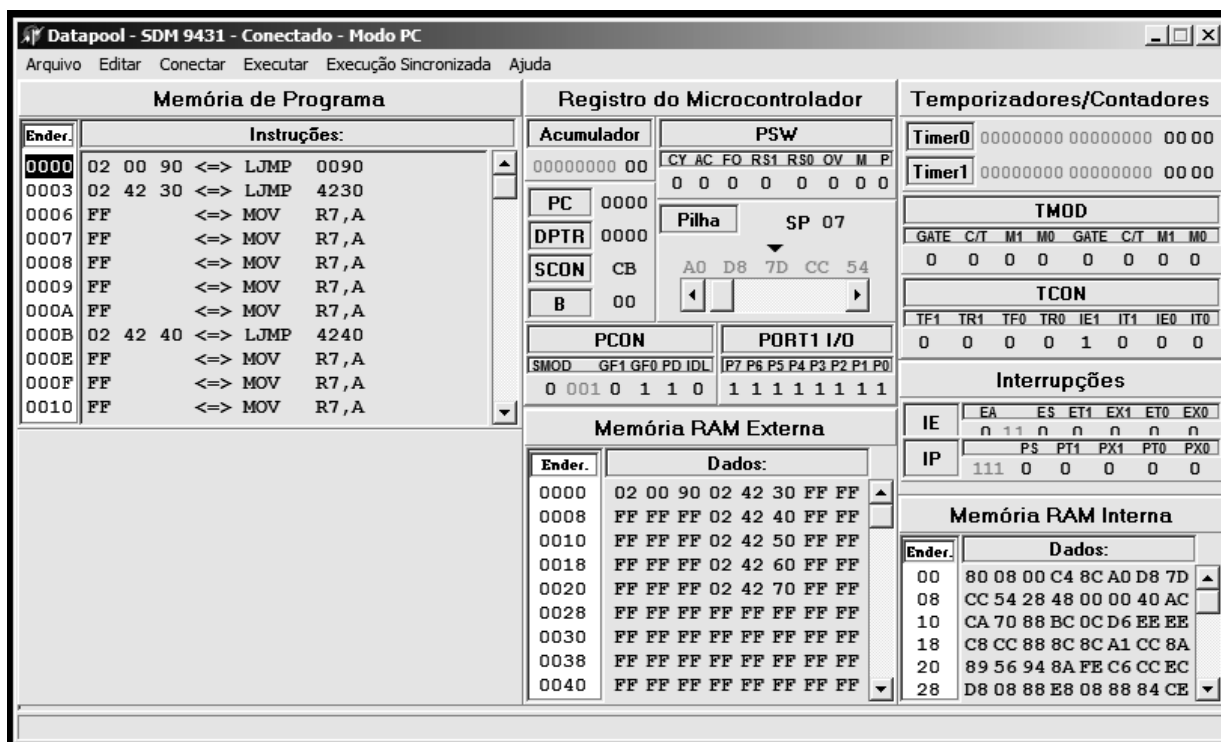



Figura 5.15 – Janela do programa SDM-9431, conectado no modo PC.

5.4 UTILIZAÇÃO DO PROGRAMA SDM-9431 NO MODO TECLADO

No modo ***TECLADO***, estão disponíveis as seguintes opções nos menus do programa SDM-9431.

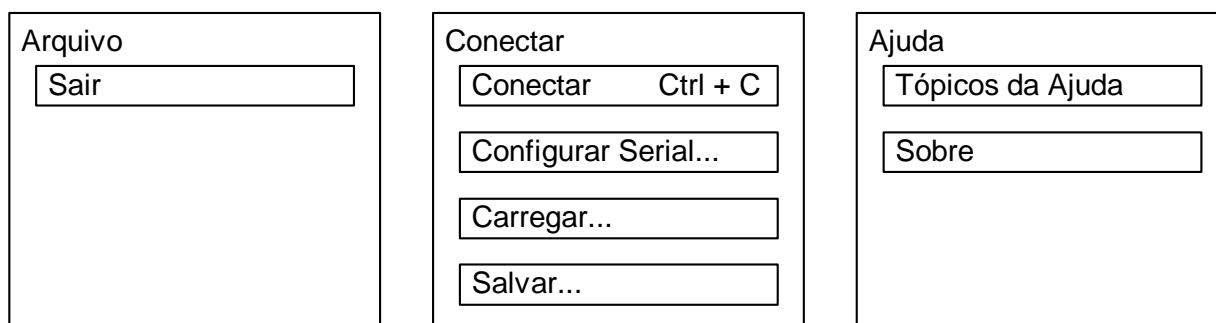



Figura 5.16 – Opções dos menus disponíveis no modo TECLADO.

5.4.1 SALVANDO ARQUIVOS DO MÓDULO NO PC

Para salvar programas da memória do Módulo como arquivo padrão HEX da Intel no PC, deve-se seguir as seguintes etapas:

1. Selecionar a opção ***Salvar...*** no menu ***Conectar***.

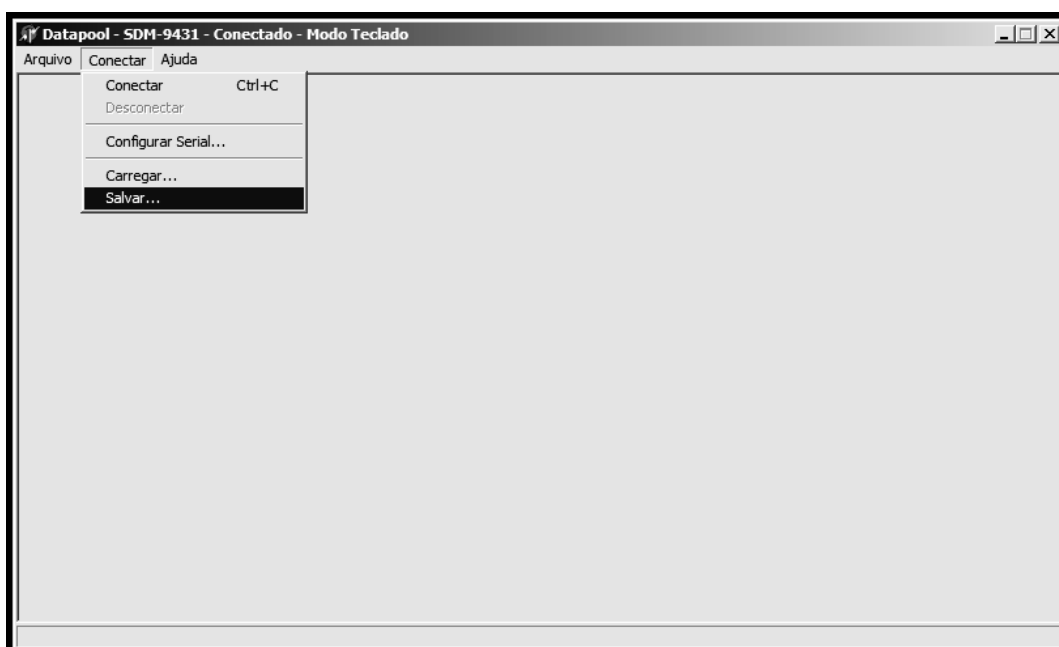



Figura 5.17 – Seleção da opção Salvar... no menu Conectar.

2. Escolher a faixa de endereço que se deseja salvar. A figura 5.18 mostra a gravação na faixa de memória de 5000h até 5500h

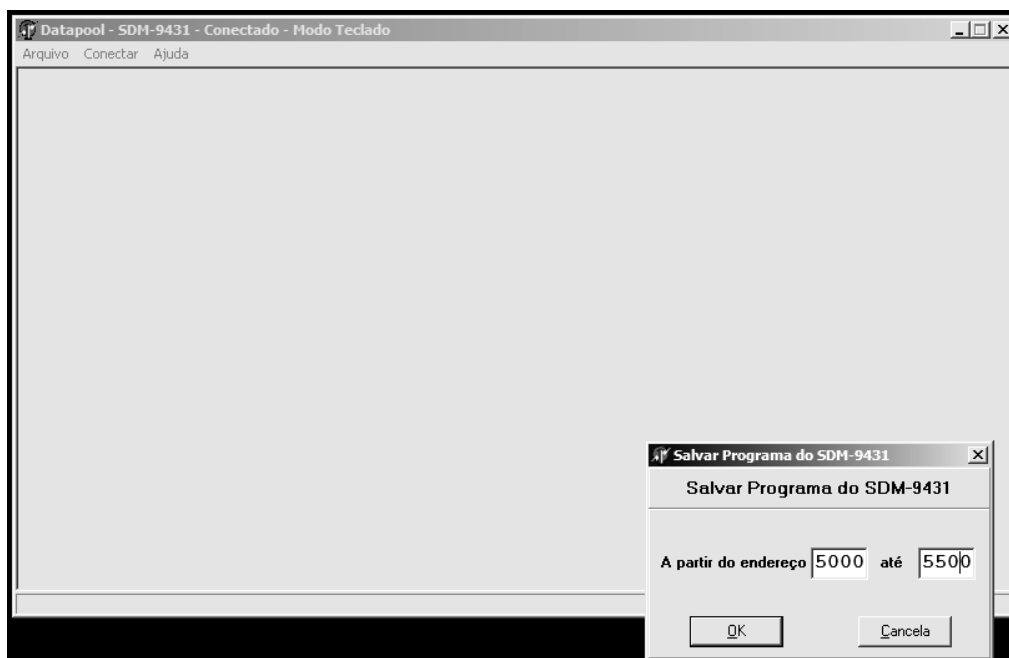



Figura 5.18 – Entrada da faixa de endereços de memória a ser salva.

3. Pressionando-se o botão ***OK*** o programa apresenta a mensagem da figura 5.19, que informa sobre o modo de se iniciar a transmissão dos dados pelo módulo.

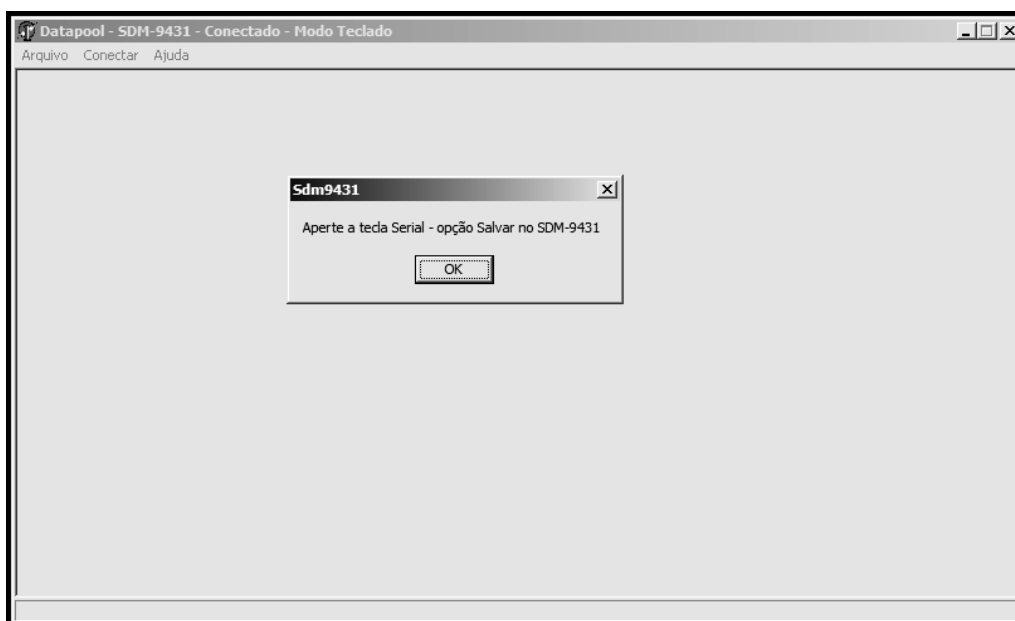



Figura 5.19 – Mensagem de instrução para iniciar a transmissão de dados.

4. A tabela seguinte mostra sequência de teclas para se iniciar a transmissão de dados do módulo para o PC.

TECLA DIGITADA	DISPLAY	COMENTÁRIO
9 SERIAL	0 - Carregar 1 - Salvar_	Após pressionar a tecla *SERIAL* deve-se escolher entre transmitir para o PC ou receber dados deste.
1 INS/VER	Aguarde_	Escolhendo a opção *SALVAR,* o módulo aguardará o PC permitir o início da transmissão.

5. Pressionando-se o botão *OK* na figura 5.19, o programa exibe uma janela para escolha da pasta e nome do arquivo, mostrada na figura 5.20.

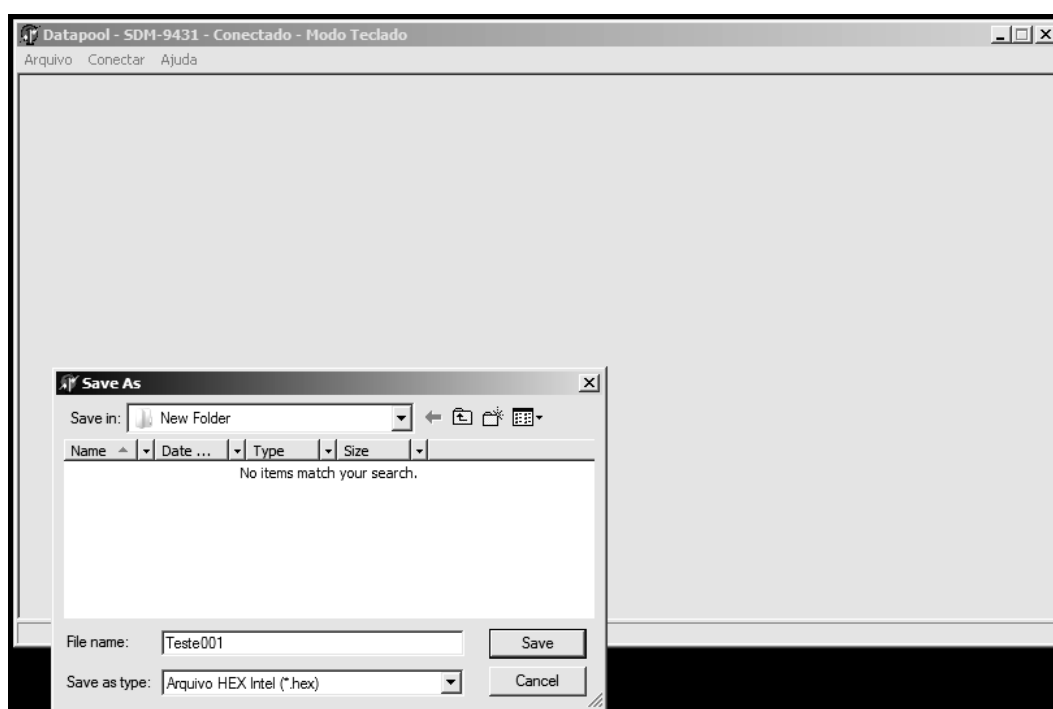



Figura 5.20 – Janela para escolha da pasta e nome do arquivo HEX.

6. Pressionando-se o botão ***Save*** o arquivo será salvo e a linha inferior da janela apresentará a mensagem ***Salvo!***, como mostra a figura 5.21. Caso haja algum erro na comunicação, será exibida a mensagem da figura 5.22.

Figura 5.21 – Mensagem indicando que a transmissão foi bem sucedida.

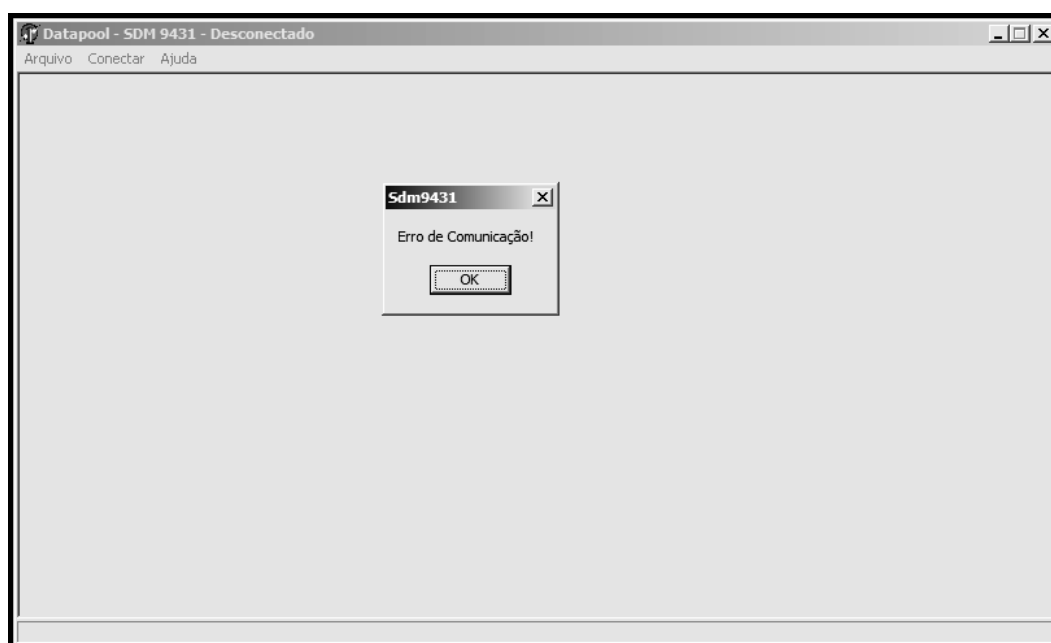



Figura 5.22 – Mensagem indicando que ocorreu erro na transmissão.

5.4.2 CARREGANDO ARQUIVOS DO PC PARA O MÓDULO

Para carregar um arquivo padrão HEX da Intel do PC como programa da memória do Módulo, deve-se seguir as seguintes etapas:

1. Selecionar a opção ***Carregar...*** no menu ***Conectar***.

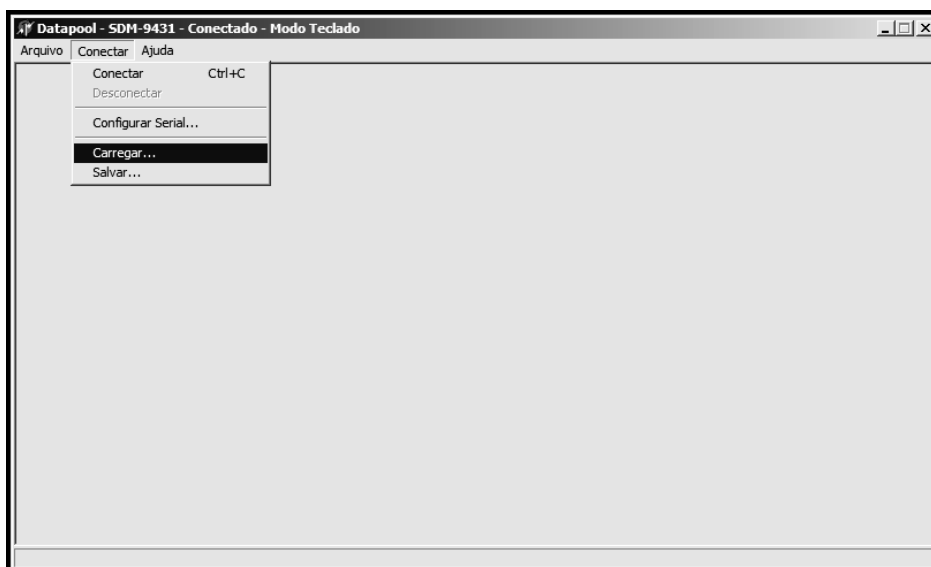



Figura 5.23 – Seleção da opção Carregar... no menu Conectar.

2. O programa exibirá uma janela para escolha da pasta e nome do arquivo a ser carregado, mostrada na figura 5.24.

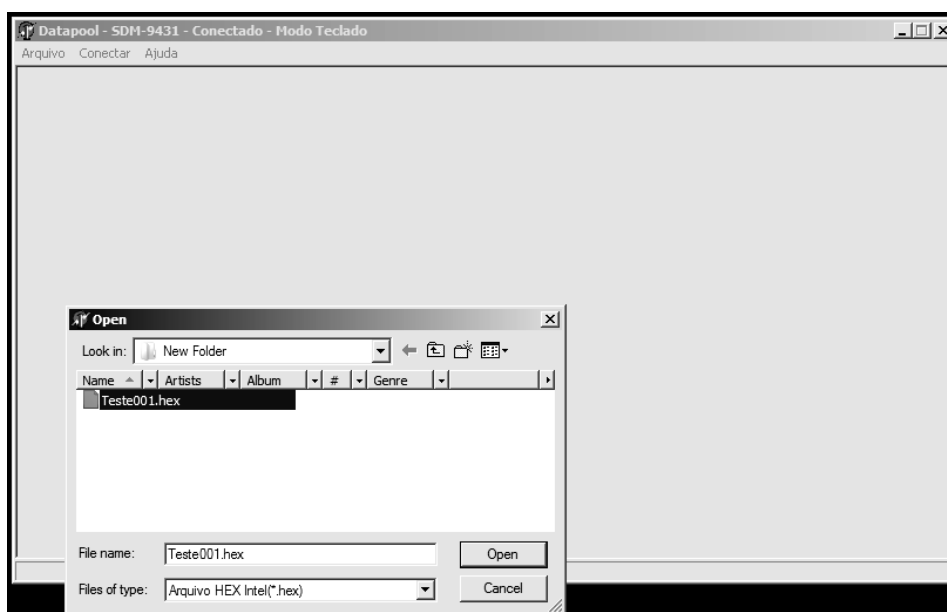



Figura 5.24 – Janela para escolha da pasta e nome do arquivo HEX.

3. Pressionando-se o botão ***Open*** o programa apresenta a mensagem da figura 5.25, que pede para que se coloque o módulo no modo de carregamento, para que dê inicio à transmissão dos dados para o módulo.

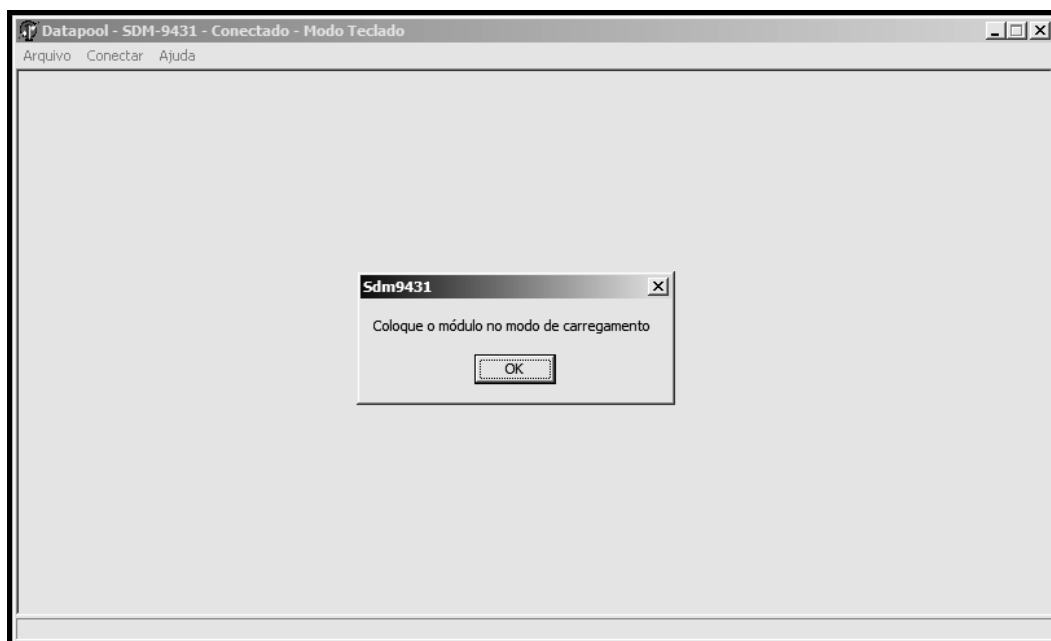



Figura 5.25 – Mensagem de instrução para iniciar a transmissão de dados.

4. A tabela seguinte mostra seqüência de teclas para se colocar o módulo SDM-9431 no modo carregamento.

TECLA DIGITADA	DISPLAY	COMENTÁRIO
9 SERIAL	0 - Carregar 1 - Salvar_	Após pressionar a tecla *SERIAL* deve-se escolher entre transmitir para o PC ou receber dados deste.
0 EXEC	Aguarde_	Escolhendo a opção *CARREGAR,* o módulo aguardará o PC permitir o início da transmissão.

5. Pressionando-se o botão ***OK*** o arquivo será carregado e será exibida a janela inicial. Caso haja algum erro na comunicação, será exibida a mensagem da figura 5.22.

5.5 UTILIZAÇÃO DO PROGRAMA SDM-9431 NO MODO PC

No modo ***PC*** estão disponíveis as seguintes opções nos menus:

Figura 5.26 – Opções dos menus disponíveis no modo TECLADO.

5.5.1 SALVANDO ARQUIVOS DO MÓDULO NO PC

Para salvar programas da memória do Módulo como arquivo padrão HEX da Intel no PC, deve-se seguir as seguintes etapas:

1. Selecionar a opção ***Salvar...*** no menu ***Conectar***.

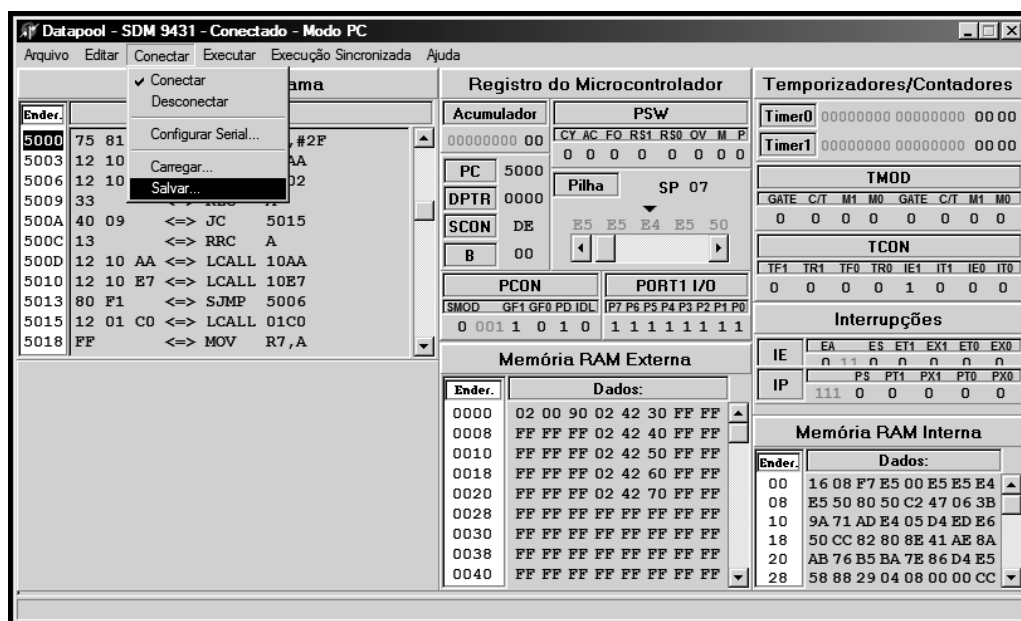



Figura 5.27 – Seleção da opção Salvar... no menu Conectar.

2. Escolher a faixa de endereço que se deseja salvar. A figura 5.28 mostra a gravação na faixa de memória de 5000h até 5500h

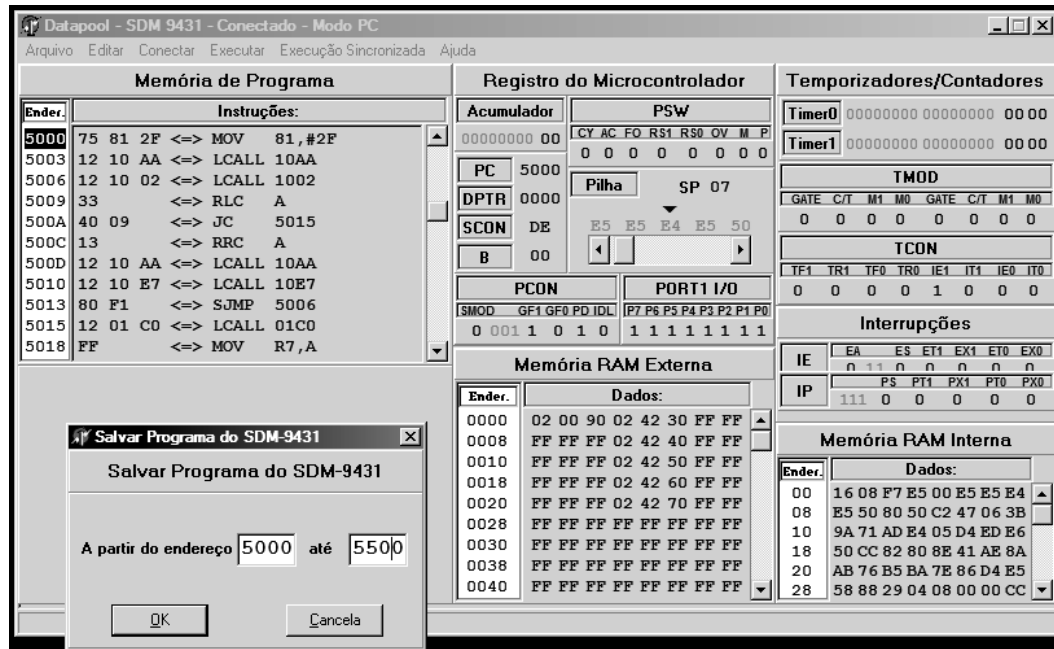



Figura 5.28 – Entrada da faixa de endereços de memória a ser salva.

3. Pressionando-se o botão *OK* na figura 5.19, o programa exibe uma janela para escolha da pasta e nome do arquivo, mostrada na figura 5.20.

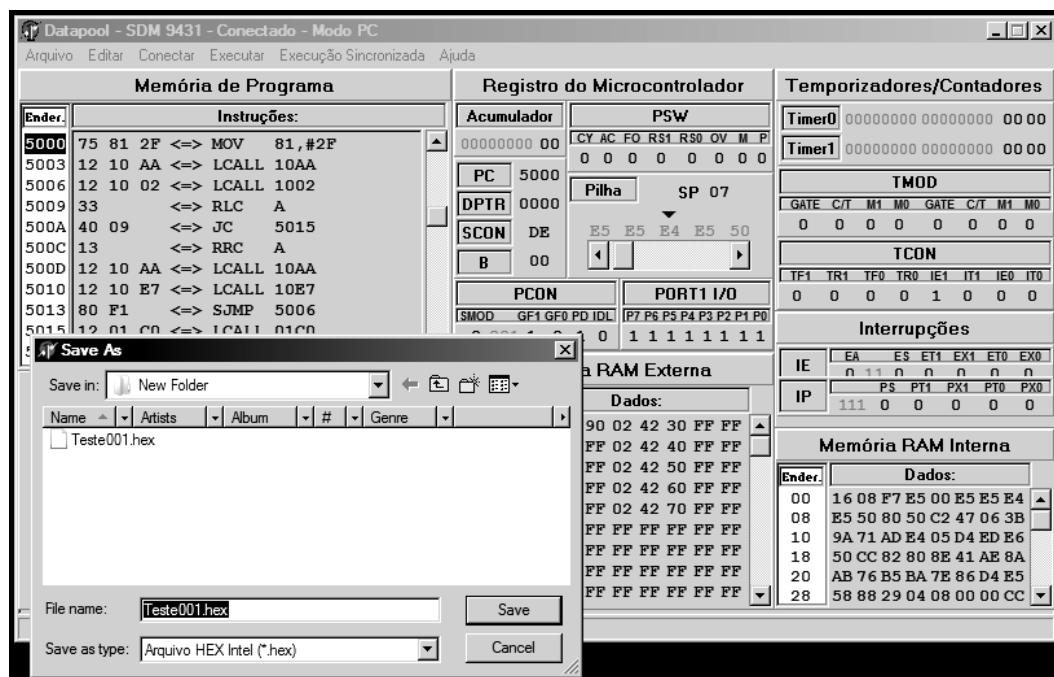



Figura 5.29 – Janela para escolha da pasta e nome do arquivo HEX.

4. Pressionando-se o botão ***Save*** o arquivo será salvo e a linha inferior da janela apresentará a mensagem ***Salvo!***, como mostra a figura 5.30. Caso haja algum erro na comunicação, será exibida a mensagem da figura 5.22.

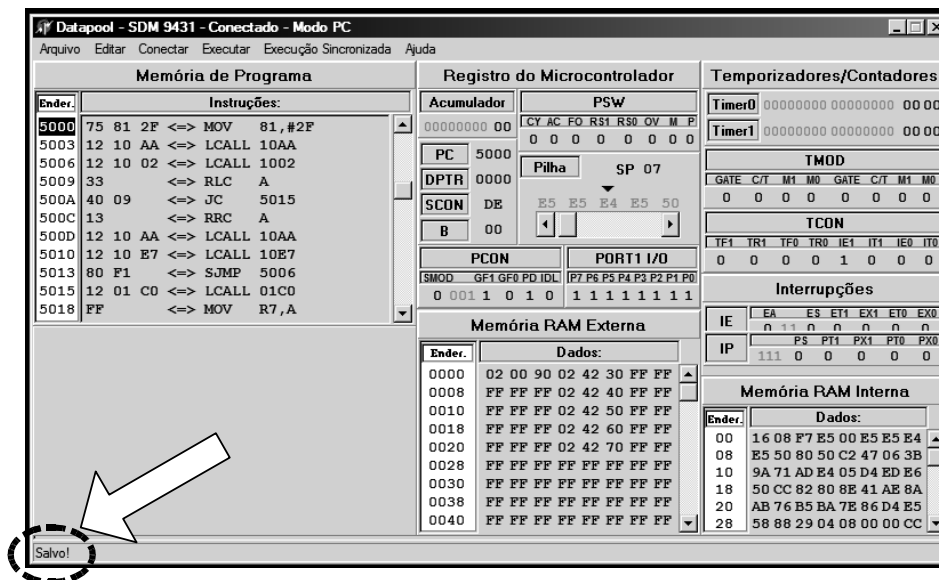



Figura 5.30 – Mensagem indicando que a transmissão foi bem sucedida.

5.5.2 CARREGANDO ARQUIVOS DO PC PARA O MÓDULO

Para carregar um arquivo padrão HEX da Intel do PC como programa da memória do Módulo, deve-se seguir as seguintes etapas:

a. Selecionar a opção ***Carregar...*** no menu ***Conectar***.

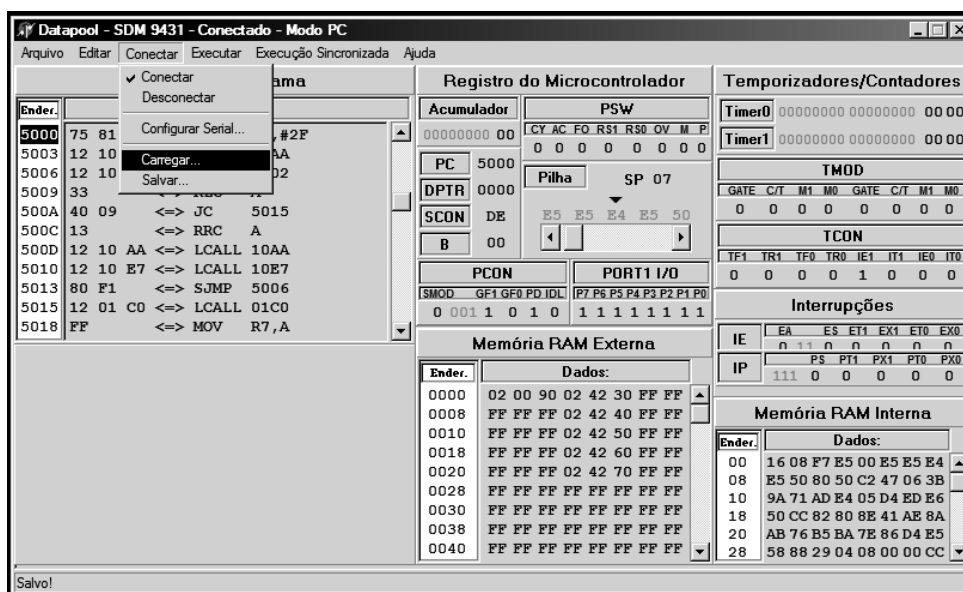



Figura 5.31 – Seleção da opção Carregar... no menu Conectar.

b. O programa exibirá uma janela para escolha da pasta e nome do arquivo a ser carregado, mostrada na figura 5.32.

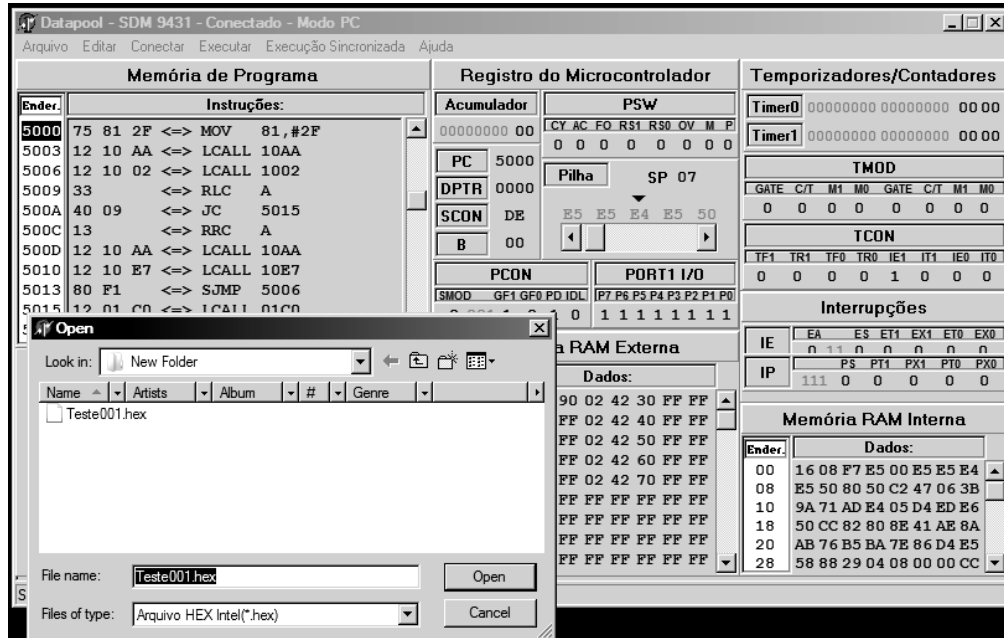



Figura 5.32 – Janela para escolha da pasta e nome do arquivo HEX.

c. Pressionando-se o botão ***Open*** o programa apresenta a tela inicial do programa SDM-9431, com o programa HEX descarregado na memória mostrado na janela ***Memória de Programa***, mostrada em destaque na figura 5.33.

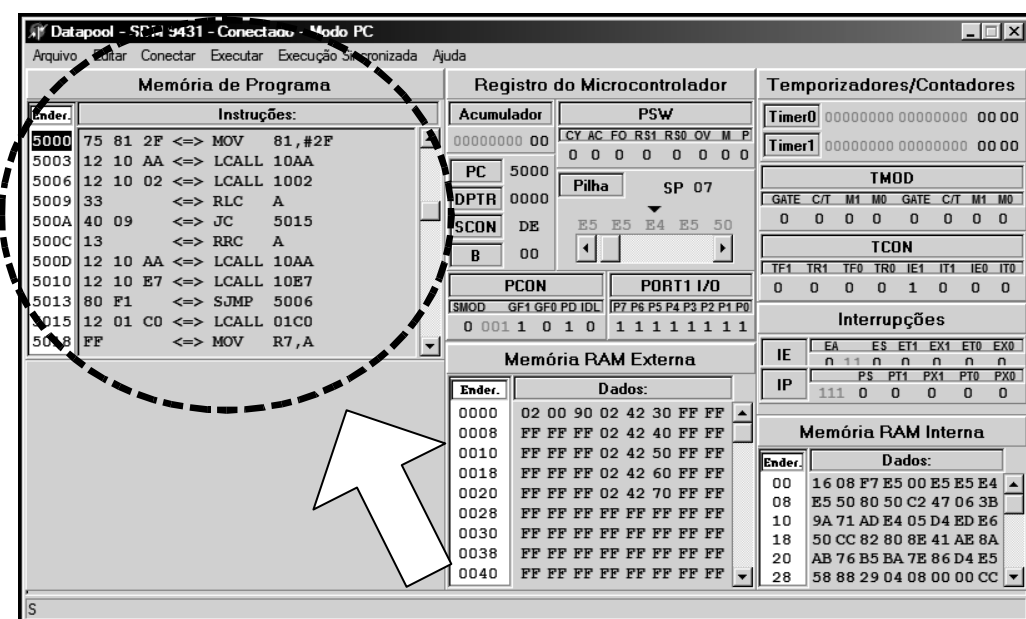



Figura 5.33 – Janela da Memória de Programa, com o programa descarregado.

5.5.3 OPERAÇÕES NA MEMÓRIA DO MÓDULO SDM-9431

O programa SDM-9431 permite realizar as operações descritas a seguir nas memórias interna e externa do microcontrolador.

- **Preencher memória**

Preenche uma área da memória (interna ou externa) do módulo SDM-9431 com um mesmo valor. Para tanto, o módulo SDM-9431 deve estar conectado no modo PC e, então, deve ser selecionada a opção ***Preencher Memória*** no menu ***Editar***, conforme mostra a figura 5.34.

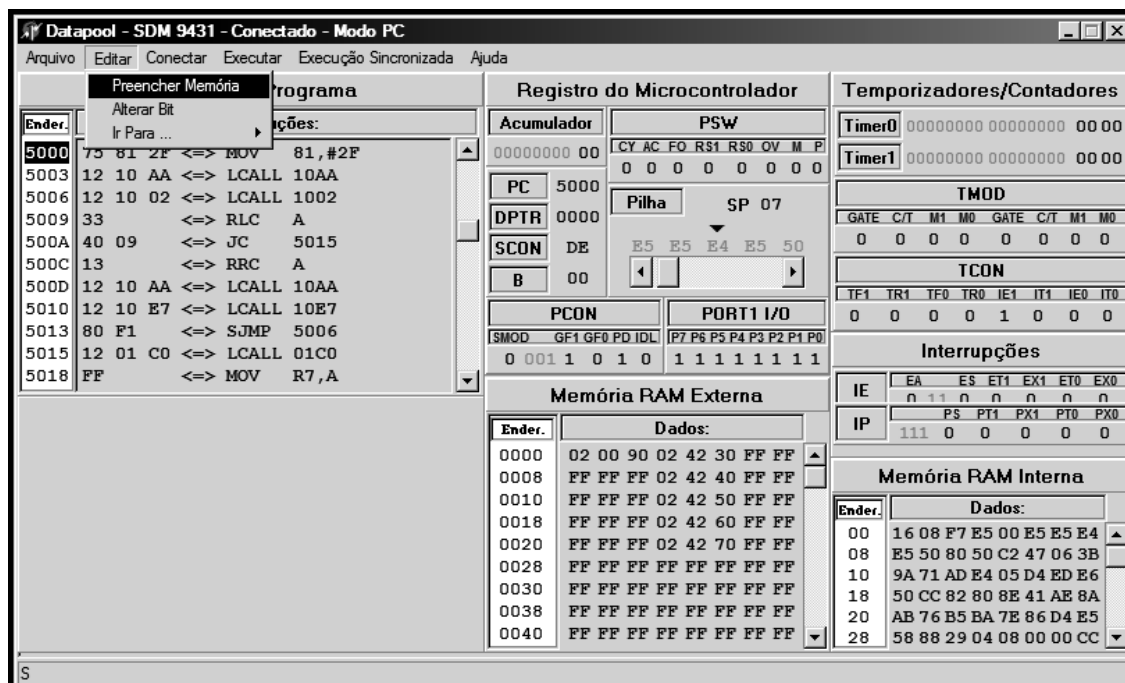



Figura 5.34 – Opção Preencher Memória no menu Editar.

Selecionando-se ***Preencher Memória***, será apresentada uma janela para escolher se a memória a ser preenchida é a externa ou a interna, a faixa de memória a ser preenchida e o valor a ser gravado na memória. A faixa de memória válida para preenchimento no módulo vai de 5000h a DFFFh.

A figura 5.35 mostra esta janela de preenchimento e a figura 5.36 mostra a área de memória preenchida com o uso desta opção.

- **Alterar bit**

Altera o valor de um bit endereçável do microcontrolador no Módulo SDM-9431. Para tanto, o módulo SDM-9431 deve estar conectado no modo PC e, então, deve ser selecionada a opção ***Alterar Bit*** no menu ***Editar***, conforme mostra a figura 5.37.

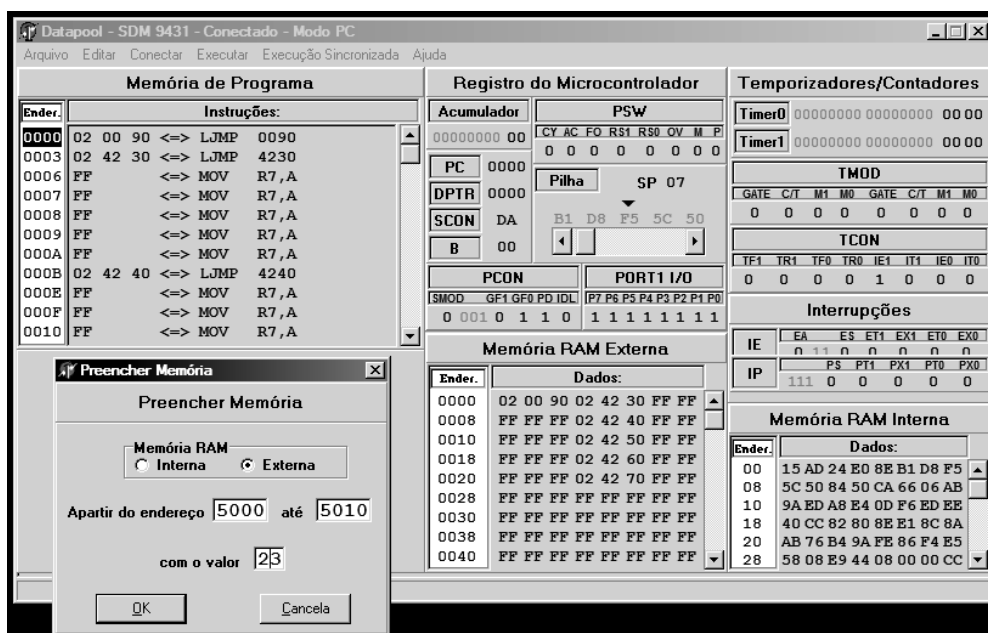



Figura 5.35 – Janela de preenchimento de memória.

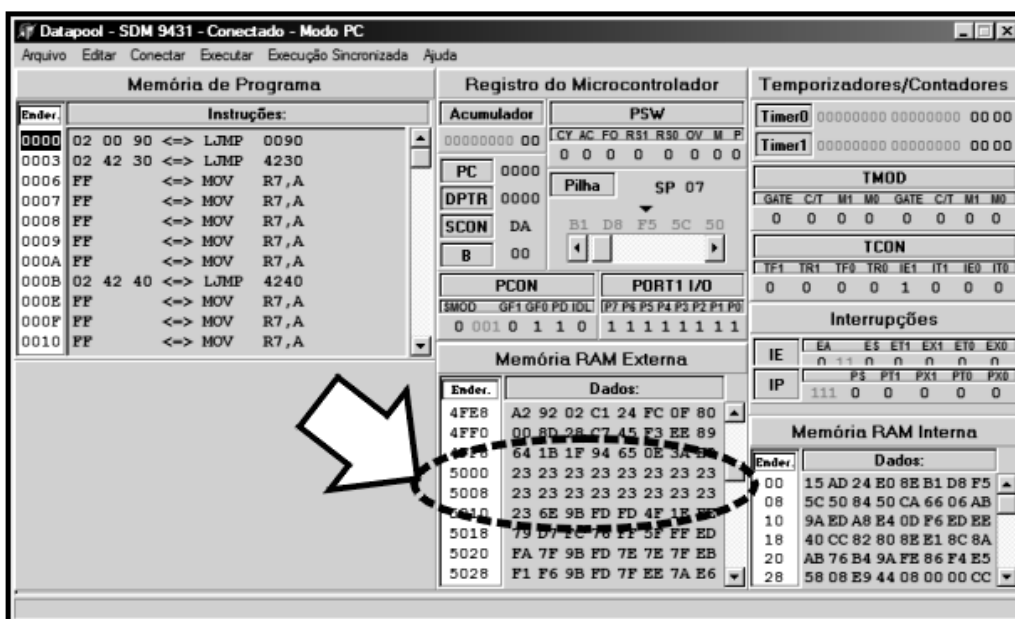



Figura 5.36 – Área de memória preenchida.

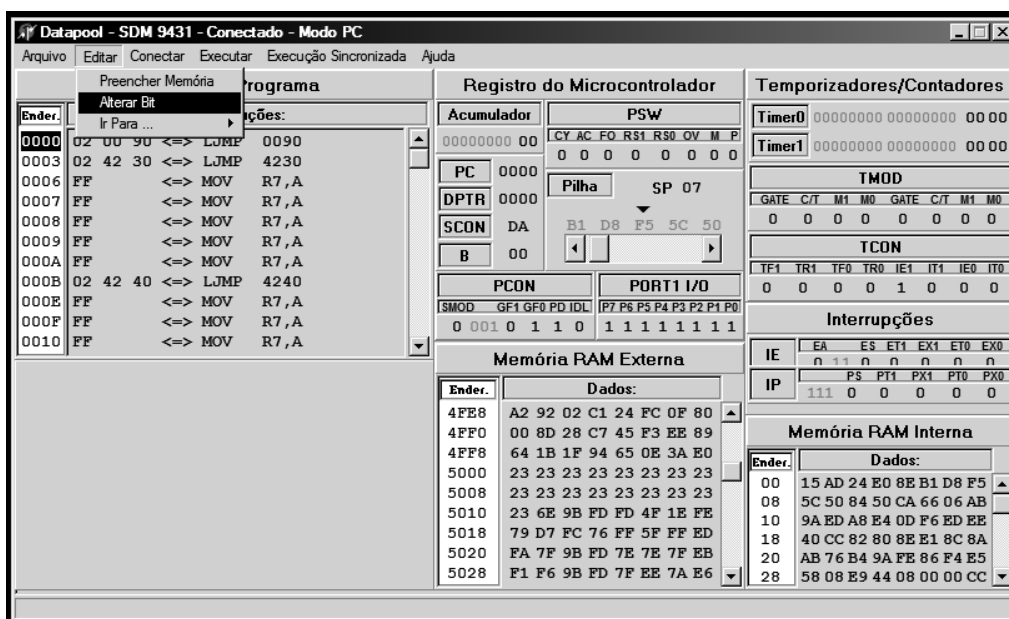



Figura 5.37 – Opção Alterar Bit no menu Editar.

Selecionando-se a opção ***Alterar Bit*** será apresentada uma janela para escolha do bit a ser alterado, mostrada na figura 5.38. Para alterar o valor do bit deve-se colocar o cursor sobre o valor mostrado e clicar com o botão esquerdo do mouse. A cada clique o valor será trocado.

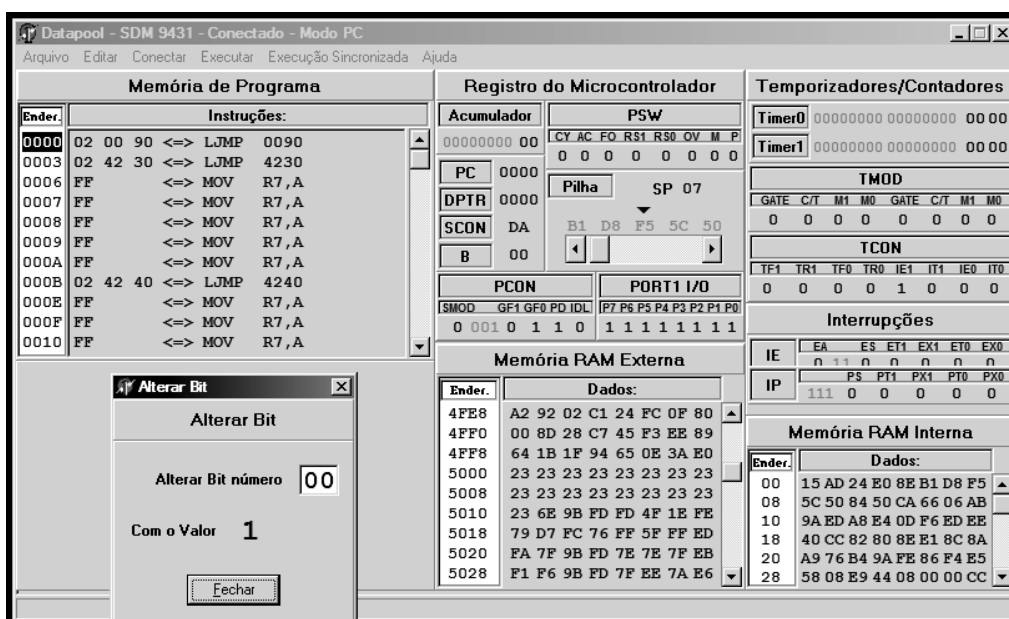



Figura 5.38 – Janela de alteração de bit.

- **Ir para um endereço específico**

Permite visualizar um endereço específico de memória do Módulo SDM-9431, seja memória de programa, RAM externa ou RAM interna. Para tanto, o módulo deve estar conectado no modo PC e, então, deve ser selecionada a opção ***Ir Para ...*** no menu ***Editar*** e, em seguida, a memória a ser visualizada, conforme mostra a figura 5.39.

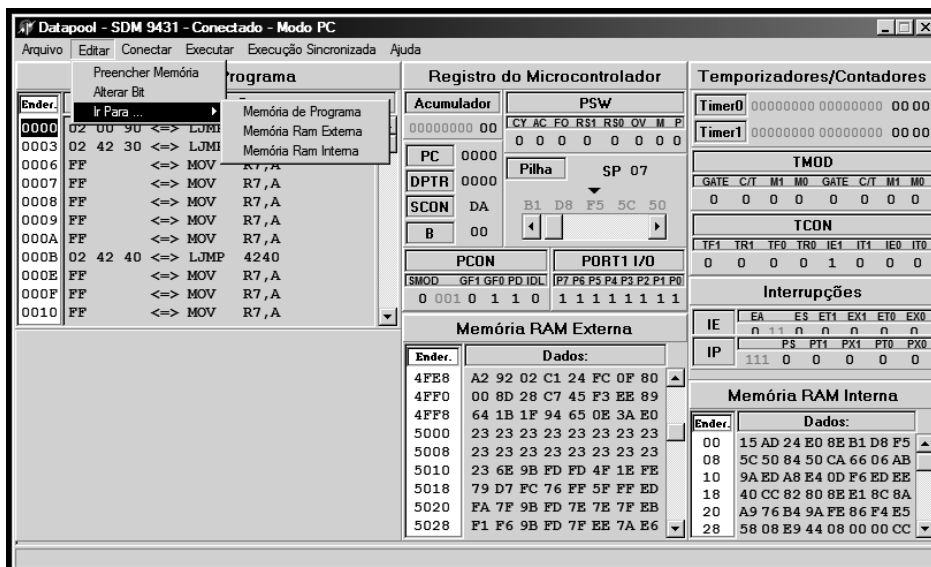



Figura 5.39 – Opção Ir Para ... no menu Editar.

Selecionando-se a memória, será apresentada uma janela para escolha do endereço a ser exibido. Nesta janela pode ser alterada a opção da memória a ser visualizada.

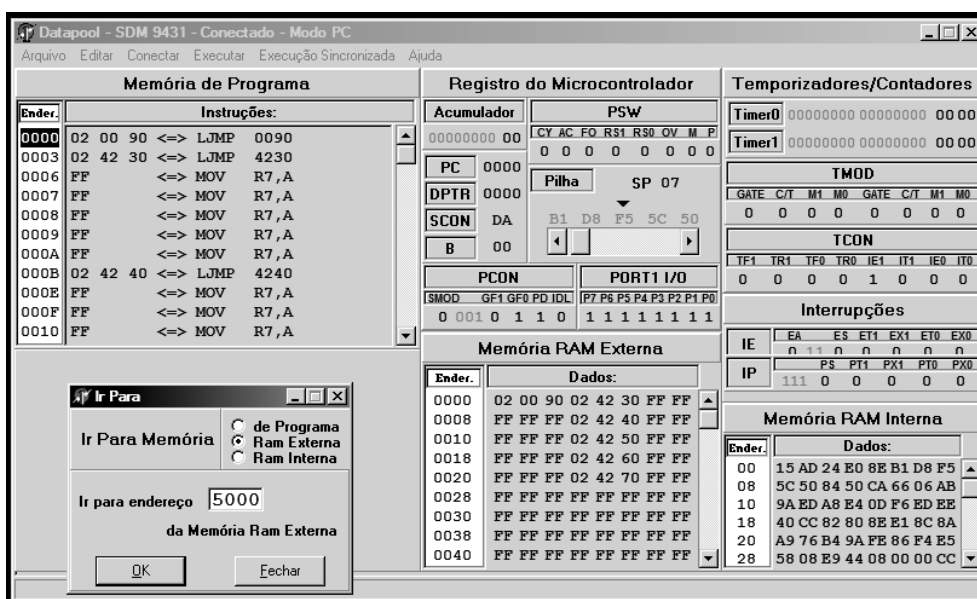



Figura 5.40 – Janela de visualização de endereço.

A figura 5.41 mostra o resultado do uso desta opção.

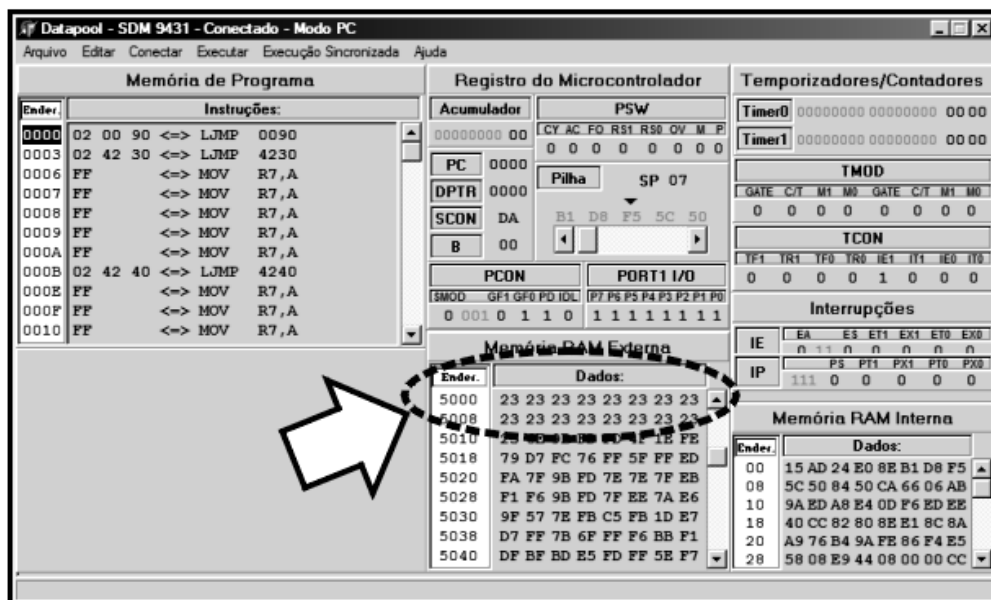



Figura 5.41 – Visualização do conteúdo do endereço selecionado.

5.5.4 OPÇÕES DO MENU EXECUTAR

As opções do menu ***Executar*** do programa SDM-9431 permitem executar programas armazenados na memória do módulo SDM-9431 em vários modos, descritos a seguir.

- **Executar**

Executa um programa armazenado na memória do módulo SDM-9431, a partir de um endereço determinado pelo usuário. Para executar um programa, o módulo SDM-9431 deve estar conectado no modo PC.

O usuário deve, então, inserir no contador de programa o endereço inicial do programa a ser executado. Para tanto, o usuário deve clicar sobre o endereço mostrado no contador de programa (***PC***) e digitar o endereço desejado, como mostra a figura 5.42.

Após digitar o endereço, a tabela de memória de programa mostra a instrução a partir da qual o programa será executado.

Finalmente, para executar o programa, deve ser selecionada a opção ***Executar*** no menu ***Executar***, conforme mostra a figura 5.43 (também pode ser usado o atalho ***CTRL+F8***).

Após esta seleção o módulo passa a executar o programa, até que o programa seja encerrado ou a tecla ***RESET*** do módulo seja pressionada (no encerramento do programa no módulo SDM-9431, pode ser necessário reconectar o módulo ao PC).

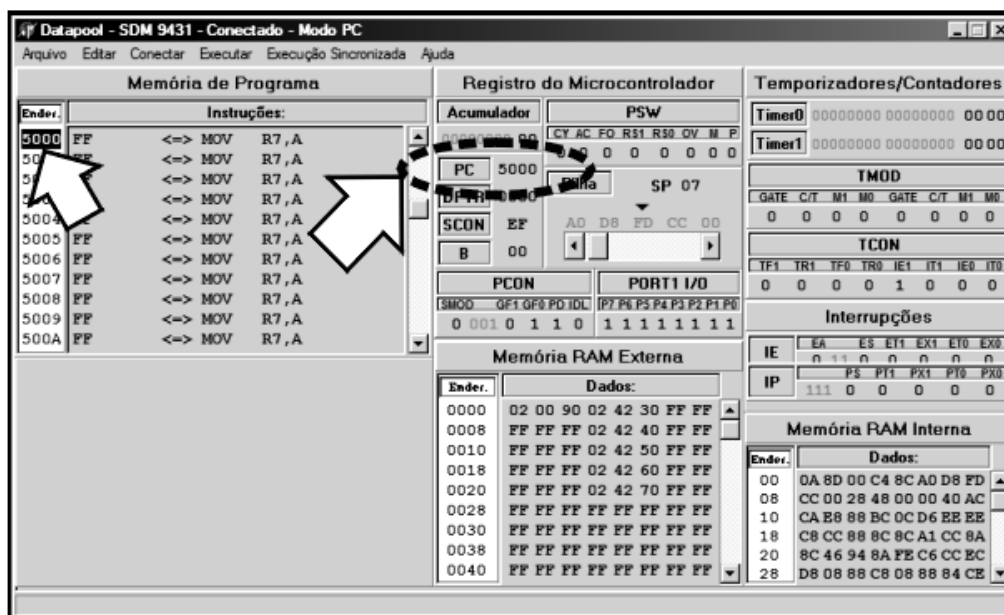



Figura 5.42 – Colocação de endereço no registro CP.

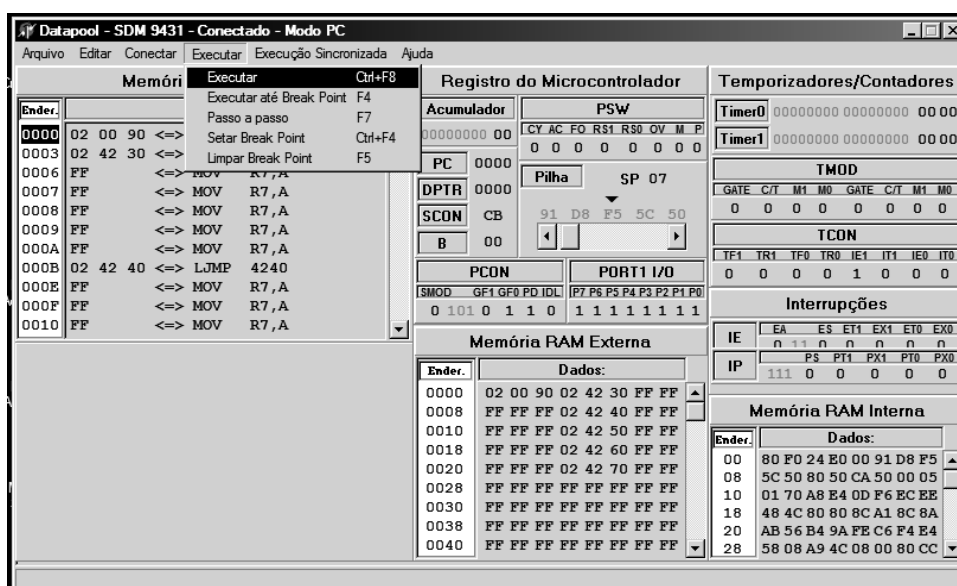



Figura 5.43 – Opção Executar do menu Executar.

- **Executar até Break Point**

Executa um programa armazenado na memória do módulo SDM-9431, a partir de um endereço determinado pelo usuário até um break point (ponto de parada) também determinado pelo usuário. Para executar um programa até um break point, o módulo SDM-9431 deve estar conectado no modo PC.

O usuário deve, então, inserir no contador de programa o endereço inicial do programa a ser executado. Para tanto, o usuário deve clicar sobre o endereço mostrado no contador de programa (***PC***) e digitar o endereço desejado, como mostra a figura 5.42.

Para determinar o break point, o usuário deve clicar sobre a instrução correspondente. A seguir, o usuário deve selecionar a opção ***Setar o Break Point*** no menu ***Executar*** (ou usar o atalho de teclado ***Ctrl+F4***) e o endereço do break point será destacado em vermelho conforme mostra a figura 5.44. Para remover o break point deve-se selecionar a opção ***Limpar Break Point*** (ou usar o atalho de teclado ***F5***).

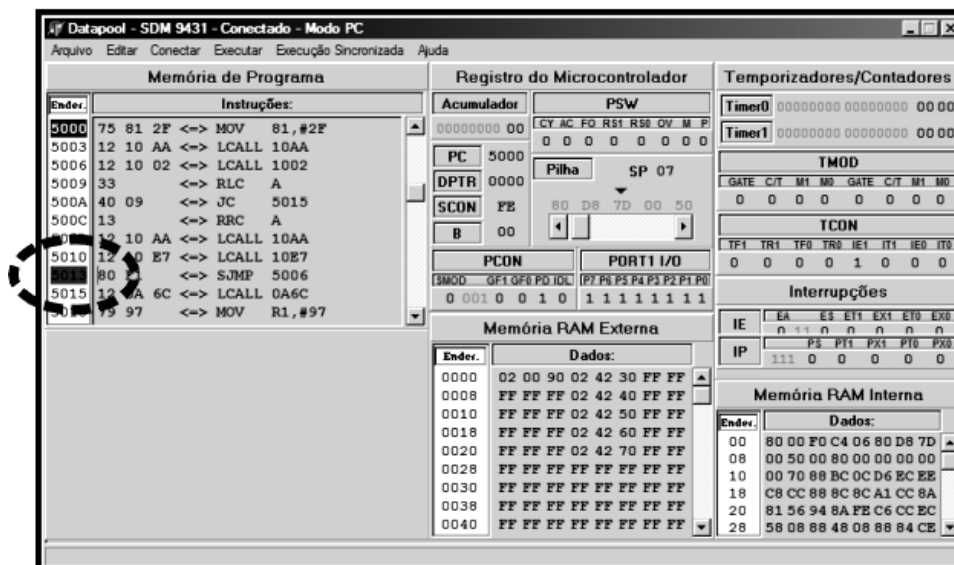



Figura 5.44 – Marcação do BREAK POINT.

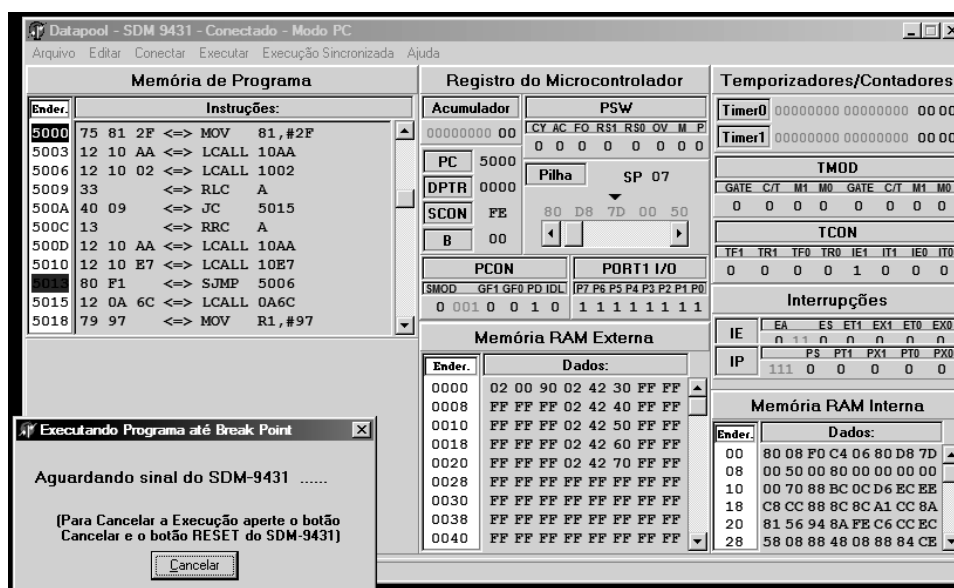



Figura 5.45 – Janela de execução de programa até o break point.

Para iniciar a execução do programa até o break point o usuário deve selecionar a opção ***Executar até Break Point*** no menu ***Executar*** (ou usar o atalho de teclado ***F4***). Após esta seleção o módulo passa a executar o programa, até o Break Point.

Durante a execução do programa será exibida uma janela, mostrada na figura 5.45. Para encerrar a execução do programa o usuário deve pressionar o botão ***Cancelar*** na janela e reconectar o módulo ao PC.

- **Executar Passo a Passo**

Executa um programa armazenado na memória do módulo SDM-9431, a partir de um endereço determinado pelo usuário, pparando a cada instrução executada. Para executar um programa no modo passo a passo, o módulo SDM-9431 deve estar conectado no modo PC.

O usuário deve, então, inserir no contador de programa o endereço inicial do programa a ser executado. Para tanto, o usuário deve clicar sobre o endereço mostrado no contador de programa (***PC***) e digitar o endereço desejado, como mostra a figura 5.42.

O usuário deve, então, selecionar a opção ***Passo a Passo***, no menu ***Executar*** (ou usar o atalho de teclado ***F7***), conforme mostra a figura 5.46.

Ao encerrar a execução do programa, pode ser necessário reconectar o módulo SDM-9431 ao PC.

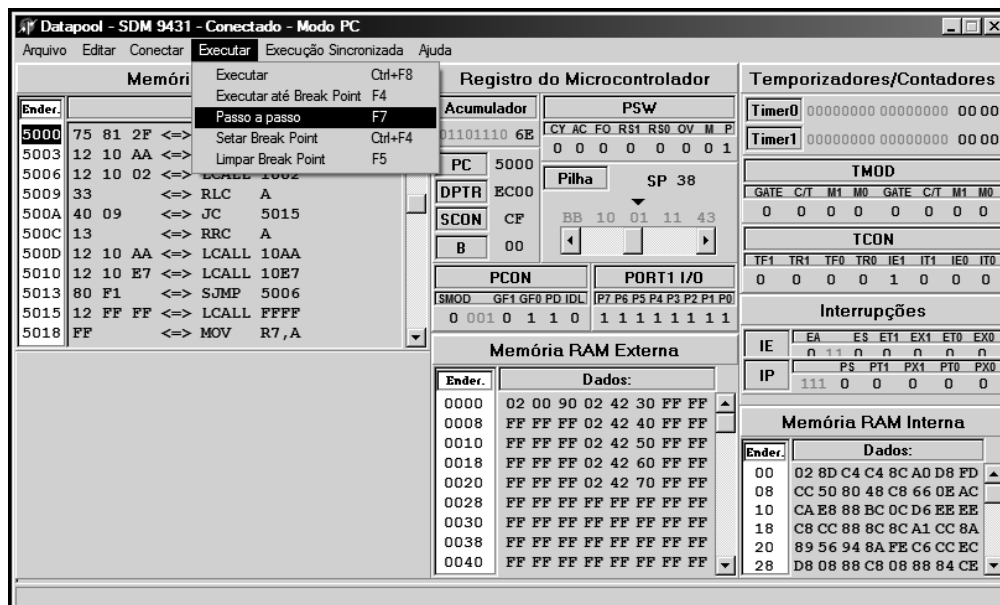



Figura 5.46 – Opção Passo a Passo do menu Executar.

Deve-se notar que programas que utilizam loops para retardo podem necessitar de um grande número de passos de execução para serem.

5.5.5 OPÇÕES DO MENU EXECUÇÃO SINCRONIZADA

As opções do menu ***Execução Sincronizada*** do programa SDM-9431 permitem executar programas armazenados na memória do módulo SDM-9431 nos modos, descritos a seguir.

- **Execução temporizada**

Executa um programa armazenado na memória do módulo SDM-9431 a partir de um endereço determinado pelo usuário, com um intervalo de tempo (determinado pelo usuário) entre as instruções. Para executar um programa no modo temporizado, o módulo SDM-9431 deve estar conectado no modo PC.

Inicialmente, o usuário deve definir o tempo de espera entre as instruções. Para tanto o usuário deve selecionar a opção ***Configurar*** no menu ***Execução Sincronizada***. Deve-se escolher, então, um dos intervalos de tempo disponíveis entre as instruções, como mostra a figura 5.47.

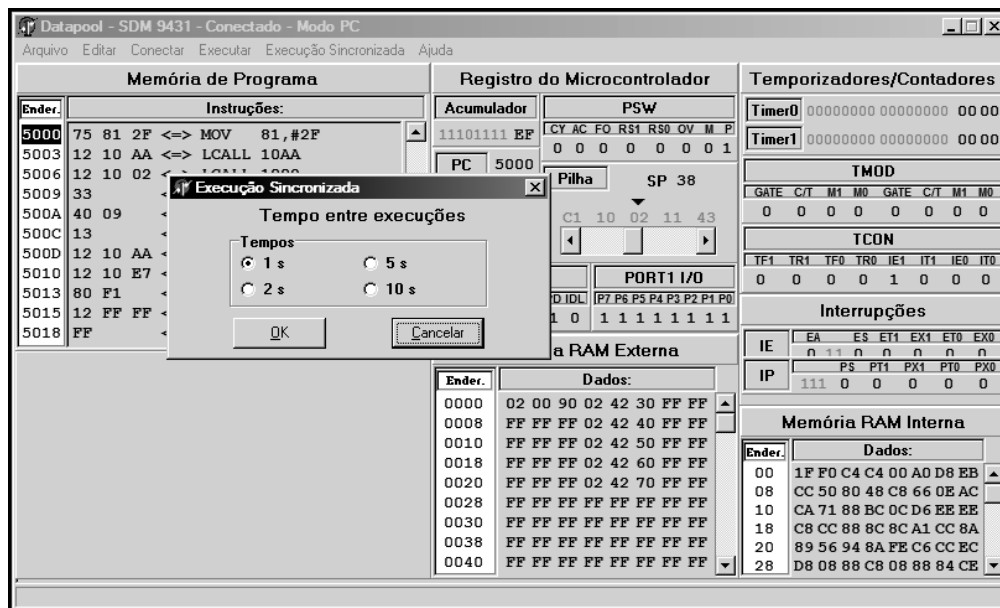



Figura 5.47 – Janela de seleção de intervalo de tempo entre instruções.

Para executar o programa, o usuário deve inserir no contador de programa o endereço inicial do programa a ser executado. Para tanto, o usuário deve clicar sobre o endereço mostrado no contador de programa (***PC***) e digitar o endereço desejado, como mostra a figura 5.42.

Para iniciar a execução temporizada o usuário deve selecionar a opção ***Execução Temporizada*** no menu ***Execução Sincronizada***. O programa passa, então, a ser executado, com um intervalo entre as instruções.

Para interromper a execução temporizada deve ser selecionada a opção ***Parar*** do menu ***Execução Sincronizada*** (durante a execução de algumas instruções o menu pode não estar disponível, então o usuário deve insistir até que o mesmo esteja disponível). Após a parada pode ser necessário reconectar o módulo SDM-9431 ao PC.

Deve-se notar que programas que utilizam a sub-rotina ***Delay*** ou implementam loops para retardo podem necessitar de um grande número de passos de execução para serem executados (255 no caso da sub-rotina ***Delay***).

- **Execução multiparada**

Executa um programa armazenado na memória do módulo SDM-9431 a partir de um endereço determinado pelo usuário, com diversos break points. Para executar um programa no modo temporizado, o módulo SDM-9431 deve estar conectado no modo PC.

O usuário deve selecionar os break points desejados. A seleção de um break point é feita colocando-se o cursor sobre o endereço desejado e pressionado o boto esquerdo do mouse.O primeiro endereço selecionado será um break-point e será identificado pela cor vermelha. Os demais endereços selecionados serão do tipo multi parada e serão identificados pela cor amarela. Para remover os break points deve-se selecionar a opção ***Limpar Break Point*** (ou usar o atalho de teclado ***F5***).

Para executar o programa, o usuário deve inserir no contador de programa o endereço inicial do programa a ser executado. Para tanto, o usuário deve clicar sobre o endereço mostrado no contador de programa (***PC***) e digitar o endereço desejado, como mostra a figura 5.42.

Para iniciar a execução temporizada o usuário deve selecionar a opção ***Execução Temporizada*** no menu ***Execução Sincronizada***. O programa passa, então, a ser executado, com um intervalo entre as instruções.

Para encerrar a execução do programa o usuário deve pressionar o botão ***Cancelar*** na janela e reconectar o módulo ao PC.

CAPÍTULO 6 – OPERAÇÃO NO MODO TECLADO VIA DOS

6.1 INTRODUÇÃO

A tecla **SERIA**L é usada para carregar, ou salvar, uma região de memória externa em uma unidade de disquete de um computador compatível com IBM-PC, conectado ao sistema SDM 9431 através da porta serial.

Este comando somente poderá ser utilizado se existir um cabo de conexão entre o módulo SDM 9431 e a porta serial do computador PC e em conjunto com o programa **SDM.EXE** desenvolvido pela DATAPOOL.

Neste caso, antes de executar esta função será necessário que já exista a comunicação entre o **Módulo SDM 9431** e o computador PC. Para tal, com o cabo serial instalado, coloque a chave de seleção de modo, localizada no canto superior esquerdo do equipamento na posição TECLADO.

Executa o programa SDM.EXE fornecido em disquete juntamente com o Módulo SDM 9431.

O computador necessário para esta operação deverá possuir um terminal de vídeo VGA, ou SVGA, e ter inserido no programa CONFIG.SYS a seguinte declaração:

DEVICE = C:\DOS\ANSI.SYS

Se esta declaração, ou declaração equivalente, não estiver incluída no CONFIG.SYS, a mesma deverá ser inserida.

O programa SDM poderá ser executado na unidade de disquete ou poderá ser copiado para um diretório do disco rígido do computador em questão. Antes de executar o programa faça uma cópia do mesmo para evitar qualquer problema de perda do programa. Entretanto, este programa, ou suas cópias, somente irá operar se o sistema SDM 9431 estiver conectado a porta serial do computador.

Tanto a porta serial 1, quanto à porta serial 2, poderão ser usadas. O programa sempre procura a porta serial 1 como padrão (DEFAULT). Para utilizar a porta serial 2 do computador o programa deverá ser executado definindo a porta 2 como unidade de comunicação, ou seja: "**SDM COM2**".

Com o sistema conectado adequadamente, a execução do programa abrirá a tela inicial apresentada na figura 6.1.

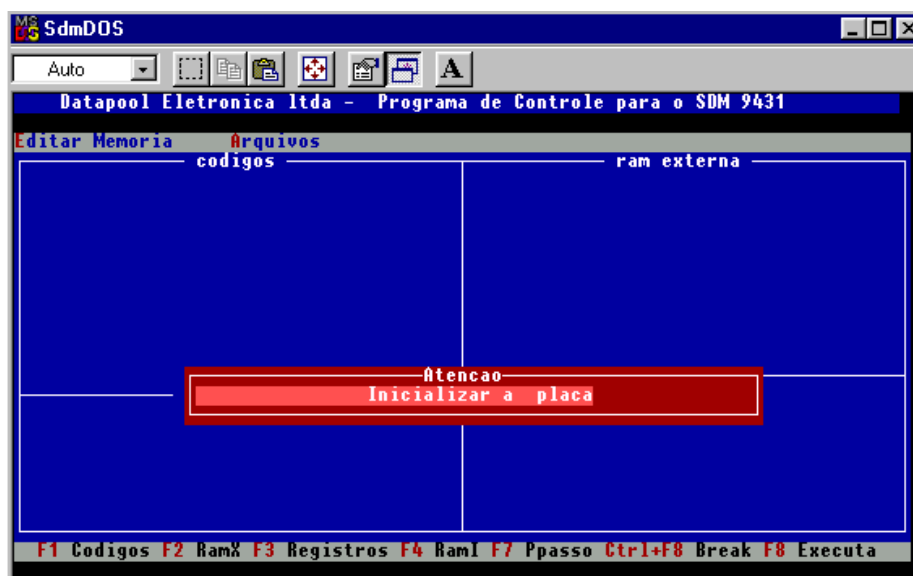



Figura 6.1 - Tela inicial.

Esta tela solicitará para que o Módulo SDM 9431 seja inicializado. Dois caminhos serão possiveis: Ao se ligar o módulo, automaticamente, o mesmo sofrerá um "reset" e, portanto, abrirá a comunicação com o software SDM. Se o módulo já estiver ligado, a inicialização deve ser feita pressionando-se a tecla "RESET" no teclado, ou o botão "RESET" posicionado no canto superior esquerdo do módulo.

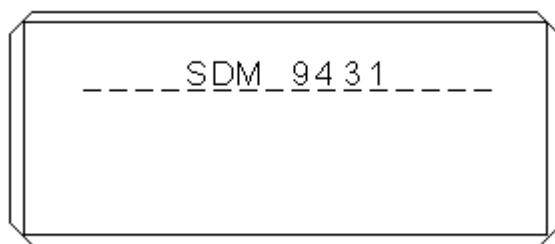
Com a comunicação completada corretamente no display do módulo aparecerá a mensagem:

No vídeo aparecerá uma tela equivalente a da figura 6.2.

Figura 6.2 - Tela de comandos gerais.

No canto superior esquerdo da tela apresentada, está em destaque a mensagem **TECLADO**. Isto significa que todo o controle do Módulo SDM 9431 será executado através de seu teclado.

Neste caso as únicas funções executáveis no programa SDM.EXE serão as de operação de arquivo, acessadas através do menu de arquivo.

O menu de arquivo é obtido quando, através das setas de movimentação (← →) for destacada a opção **"Arquivo"** e pressionado **ENTER**, ou então, através da letra **"A"** em destaque na tela. Assim será apresentada a tela da figura 6.3.

Figura 6.3 - Tela de comandos de arquivo para o modo teclado.

O menu apresentado terá as opções **carregar, salvar e fim (DOS).**

A tecla **ESC** será usada para retornar ao menu de comandos gerais.

A opção **carregar** será usada para carregar um programa no Módulo SDM 9431, através da comunicação serial.

Selecionando-se esta opção será apresentada a tela da figura 6.4.

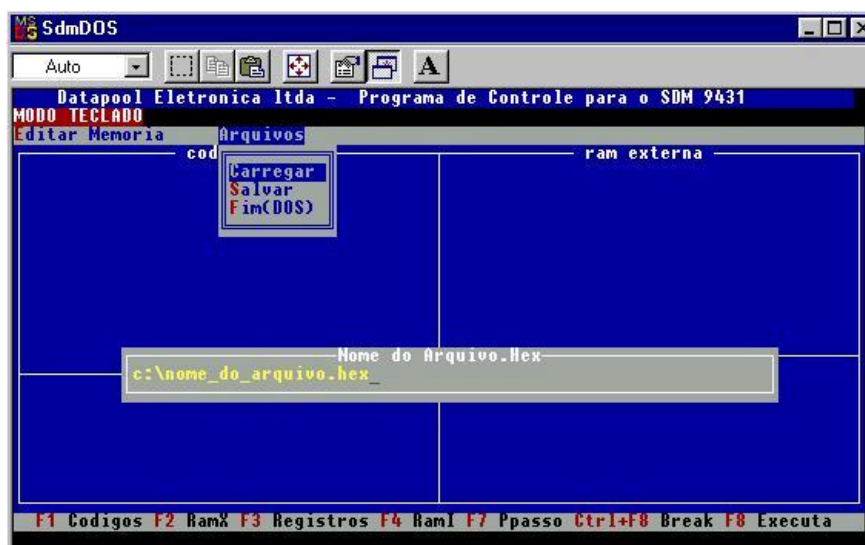



Figura 6.4 - Tela para opção carregar no modo teclado.

Uma janela de entrada para o nome do arquivo será aberta.

No teclado do Módulo SDM 9431 pressione a tecla **SERIAL**. O display apresentará a opção **carregar** (0), ou **salvar** (1), conforme a figura 5.

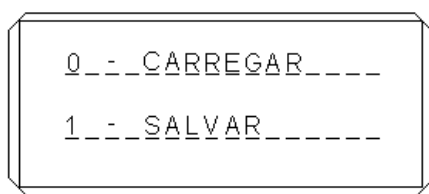



Figura 6.5 - Display para as operações da função serial.

Selecione a operação carregar (0). Uma mensagem de **"Aguarde"** será apresentada. Neste ponto o módulo estará esperando pelo recebimento do arquivo armazenado em disquete.

No computador PC especifique o arquivo.hex que se deseja carregar, incluindo o diretório e subdiretórios do caminho onde este arquivo está armazenado.

Com a transferência realizada o computador PC retorna ao menu de comandos gerais e o Módulo SDM 9431 retorna ao programa monitor.

O programa carregado estará armazenado nos endereços definidos pelo X do arquivo e, portanto, poderá ser executado.

Para salvar uma região de memória externa em uma unidade de disquete, selecione a opção **Salvar** do menu de arquivos do programa SDM.EXE.

A tela apresentada será equivalente a da figura 6.6.

Figura 6.6 - Tela para opção salvar no modo teclado.

Uma janela de entrada será aberta. Entre com o endereço inicial da região a ser salva, com o endereço final da região e com o nome do arquivo a ser criado, juntamente com o caminho de acesso deste arquivo. Deverá haver um espaço separando estes valores.

No teclado do Módulo SDM 9431 pressione a tecla SERIAL e selecione a opção SALVAR. A região de memória, especificada pelos endereços inicial e final, será transferida para o disquete no formato .HEX e com o nome atribuído para o arquivo.

6.2 OPERAÇÃO NO MODO PC

A operação no **MODO PC** é realizada através do posicionamento da chave de seleção de modo (canto superior esquerdo do equipamento) na posição PC da conexão do Módulo SDM 9431 com a porta serial do computador compatível com IBM-PC e da execução do programa SDM.EXE desenvolvido pela DATAPOOL.

O computador necessário para esta operação deverá possuir um terminal de vídeo VGA, ou SVGA, e ter inserido no programa CONFIG.SYS a seguinte declaração:

DEVICE = C:\DOS\ANSI.SYS

Se esta declaração, ou declaração equivalente, não estiver incluída no CONFIG.SYS, a mesma deverá ser inserida.

O programa SDM poderá ser executado na unidade de disquete ou poderá ser copiado para um diretório do disco rígido do computador em questão. Antes de executar o programa faça uma cópia do mesmo para evitar qualquer problema de perda do programa. Entretanto, este programa, ou suas cópias, somente irá operar se o sistema SDM 9431 estiver conectado a porta serial do computador.

Tanto a porta serial 1, quanto à porta serial 2, poderão ser usadas. O programa sempre procura a porta serial 1 como padrão (DEFAULT). Para utilizar a porta serial 2 do computador o programa deverá ser executado definindo a porta 2 como unidade de comunicação, ou seja: **"SDM COM2"**.

- ⇨ Conecte o Módulo SDM 9431 ao computador PC.
- ⇨ Selecione o modo de operação PC.
- ⇨ Execute o programa SDM.EXE através do disquete, ou de um diretório do disco rígido.

6.2.1 COMANDOS DE INICIALIZAÇÃO

A execução abrirá a tela inicial do programa apresentada na figura 6.7.

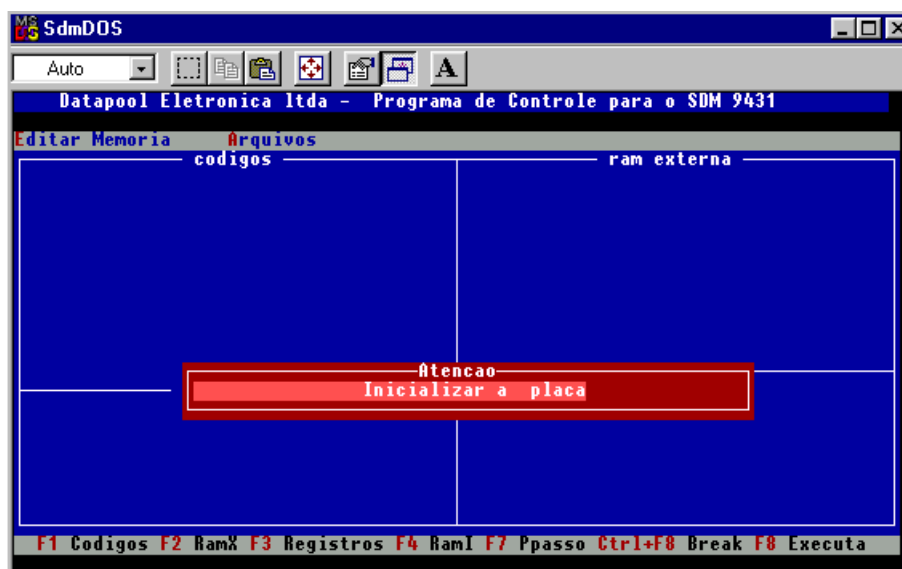



Figura 6.7 - Tela inicial.

Esta tela solicitará para que o Módulo SDM 9431 seja inicializado. Dois caminhos serão possíveis: Ao se ligar o módulo, automaticamente, o mesmo sofrerá um "reset" e, portanto, abrirá a comunicação com o software SDM. Se o módulo já estiver ligado, a inicialização deve ser feita pressionando-se a tecla "RESET" no teclado, ou o botão "RESET" posicionado no canto superior esquerdo do módulo.

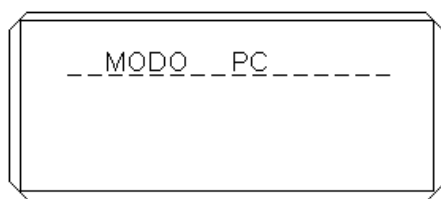
Com a comunicação completada corretamente no display do módulo aparecerá a mensagem:

No vídeo aparecerá uma tela equivalente a da figura 6.8. Os valores apresentados nas áreas de RAM serão aleatórios.

Figura 6.8 - Tela de comandos gerais.

No canto superior esquerdo da tela apresentada, está em destaque a mensagem **MODO PC**. Isto significa que todo o controle do Módulo SDM 9431 será executado através do computador PC.

A tela está dividida em quatro campos de funções, em uma linha superior de comandos, onde se tem às opções **Editar Memória** e **Arquivos**, e uma linha inferior de comandos, onde se tem as opções **F1, F2, F3, F4, F7, Ctrl + F8 e F8**.

Os campos de funções são denominados:

- Campo de códigos
- Campo de ram externa
- Campo de ram interna
- Campo de registros.

O **campo de códigos** é destinado a apresentação dos conteúdos dos endereços especificados. Na primeira coluna aparece o endereço da posição de memória em destaque, na segunda coluna aparece o opcode da instrução deste endereço, na terceira e quarta colunas, as informações complementares da instrução, quando existir, sendo dados imediatos, deslocamentos relativos, endereços, etc. Na quinta coluna tem-se o mnemônico da instrução acompanhado do correspondente endereço, quando houver.

O **campo de registros** apresenta os conteúdos dos correspondentes registros de função especial da família 8051. O registro PSW teve os seus bits de "flag" apresentados neste campo, são eles: o bit de carry (CY), carry auxiliar (AC), bits de seleção de banco de registros (RS1 e RS0) e o bit de overflow (OV). O flag de paridade é o bit menos

significativo, portanto se o valor apresentado no registro PSW for ímpar, então P = 1 ou seja, paridade ímpar, se o valor for P = 0, ou seja paridade par.

O **campo de ram externa** apresenta os conteúdos dos endereços de RAM externa (de 0000H até FFFFH), a partir de um endereço inicial selecionado pelo usuário.

O **campo de ram interna** apresenta os conteúdos dos endereços de RAM interna (de 00H até 7FH), a partir de um endereço inicial selecionado pelo usuário.

A seleção de um comando desta tela poderá ser realizada através da letra correspondente em destaque ou da tecla **ENTER**, quando a palavra de comando estiver em destaque. Com as setas de deslocamento (→↑↓←) do teclado poderá ser destacada a palavra de comando. A tecla ESC encerra qualquer comando indesejável.

Inicialmente serão apresentados os comando relativos ao manuseio de arquivos.

6.2.2 COMANDO DE ARQUIVOS

Ao ser selecionado os comando de arquivo será apresentada à tela da figura 6.9.

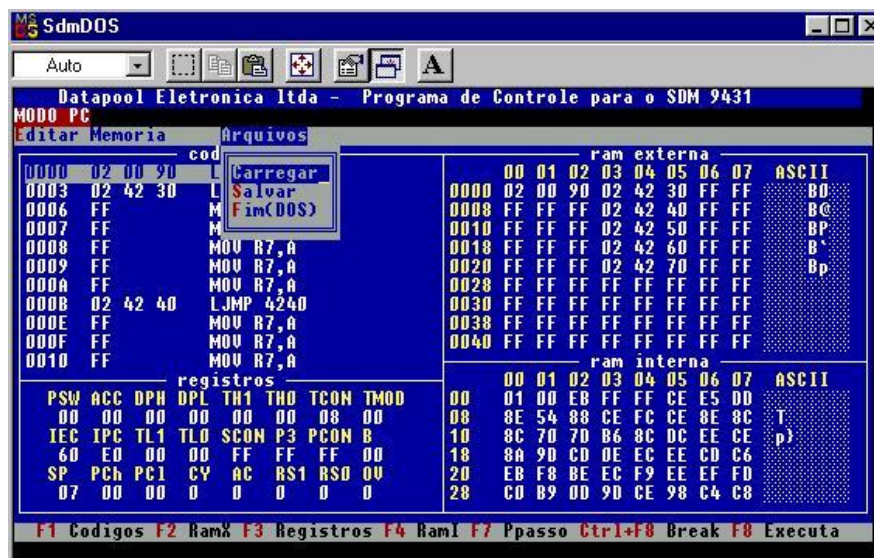



Figura 6.9 - Tela de comandos de arquivo.

Um novo menu será apresentado, tendo as opções **Carregar, Salvar e FIM (DOS).**

Observação:

Se por algum uso errôneo do teclado o programa ficar travado o CTRL + C poderá ser usado para abortar e retornar ao sistema DOS.

O comando **Carregar** será usado para buscar um arquivo no formato Intel (HEX) e colocá-lo na correspondente área de memória.

No disquete que acompanha o programa SDM tem-se um subdiretório denominado \HEX com alguns programas demonstrativos os quais serão analisados posteriormente.

- ⇨ Selecione o comando Carregar e pressione a tecla ENTER.
- ⇨ Será apresentada a tela da figura 6.10.

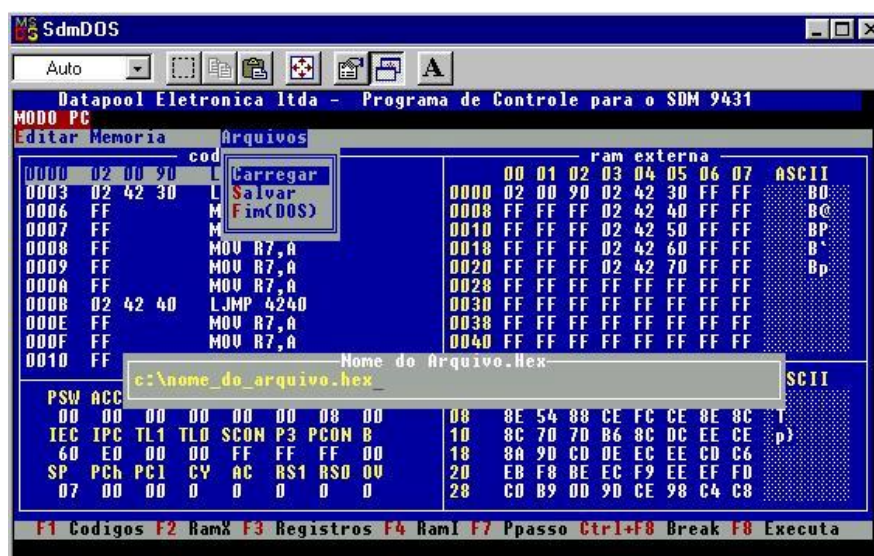



Figura 6.10 - Tela de carregamento de arquivo.

Uma nova janela solicitando a especificação de um arquivo será aberta. Somente arquivos no padrão HEX poderão ser especificados. Como exemplificação carregue o programa denominado CONT_DEC.HEX do disquete em anexo. Para tal, deve ser especificado o caminho correto de acesso deste programa, ou seja, se o disquete estiver no drive B, então digite: **B:\HEX\Cont_dec.hex** e posteriormente a tecla **ENTER**.

Antes de continuarmos analisando os comandos de arquivo, passaremos a analisar os comandos de Editar Memória. Posteriormente veremos as outras opções do menu de arquivos, conforme apresentado na figura 6.19.

6.2.3 COMANDOS DE EDITAR MEMÓRIA

O acesso ao menu de Editar Memória é feito ao pressionar a tecla **ENTER**, quando no menu de comandos gerais estiver em destaque a opção **Editar Memória**, ou diretamente pressionando-se a letra correspondente em destaque que ativa esta opção.

Ao ser selecionada esta opção será apresenta a tela da figura 6.11.

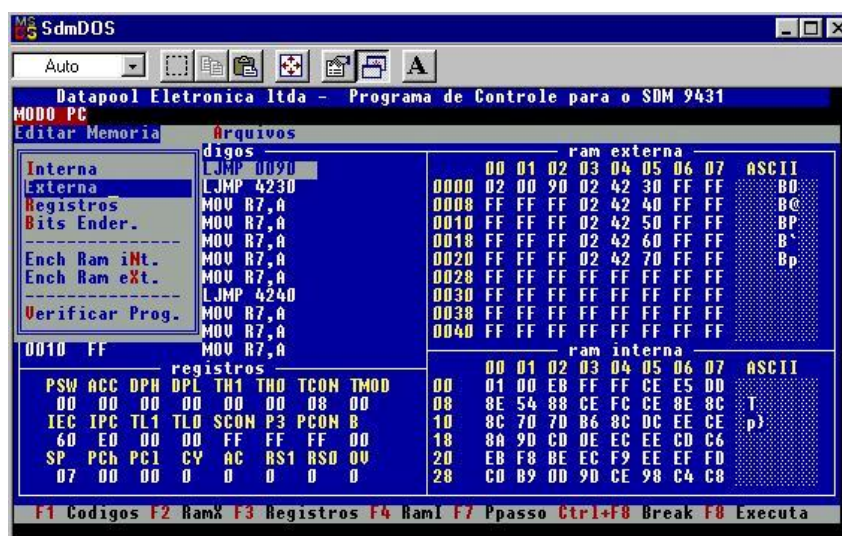



Figura 6.11 - Tela de comandos de editar memória.

Um novo menu será aberto, contendo as seguintes opções: **Interna, Externa, Registros, Bits Ender., Ench Ram iNt., Ench Ram eXt. e Verificar Prog.**

A opção **Interna** é usada para acessar o campo de ram interna, para modificação da região de endereços a ser destacada, ou para a edição de novos dados da região em destaque. A tecla **ESC** retorna ao menu de comandos gerais.

A opção **Externa** é usada para acessar o campo de ram externa para modificação da região de endereços a ser destacada, ou edição de novos dados da região em destaque.

A tecla **ESC** retorna ao menu de comandos gerais.

A opção **Registros** é usada para acessar o campo de registros para alterar o conteúdo de um registro específico.

A opção **Bits ender**. é usada para acessar ou modificar o conteúdo de um bit endereçável da família 8051.

A opção **Ench RAM iNt**. é usada para encher uma região de ram interna, desde o endereço inicial até o endereço final, com um dado específico.

A opção **Ench RAM eXt**. é usada para encher uma região de ram externa, desde o endereço inicial até o endereço final, com um dado específico.

A opção **Verificar Prog.** é usada para acessar o campo de códigos para apresentar o conteúdo da região de endereços em destaque no formato mnemônico de cada instrução.

Observação:

Uma edição no campo de ram externa irá automaticamente alterar o campo de códigos se ambos os campos estiverem destacando a mesma região de memória.

A alteração do valor do conteúdo do contador de programa PCH e PCL automaticamente altera o campo de códigos.

- ⇨ Selecione a opção Externa do menu de editar memória.
- ⇨ Será apresentada a tela da figura 6.12.

Figura 6.12 - Opção externa do menu de editar memória.

- ⇨ Uma janela de solicitação de endereço será aberta.

Entre com o endereço **5000** e pressione **ENTER**. No campo de ram externa estará em destaque a região de endereços iniciada pelo endereço 5000H. O cursor está posicionado neste campo para possibilitar alterações dos dados de um endereço específico. Através das setas de movimentação e dos valores hexadecimais (0 até 9 e A até F) pode ser alterado o conteúdo de um endereço selecionado.

Os valores apresentados para a região de memória são os correspondentes ao programa Cont_Dec.Hex, armazenado pela opção **carregar** do menu de arquivos.

- ➩ Para retornar ao menu de comandos gerais pressione a tecla ESC.
- ➩ Selecione a opção Verificar Prog. do menu de Editar Memória.
- ➩ Será apresentada a tela da figura 6.13.

Figura 6.13 - Opção verificar prog. do menu de editar memória.

Uma janela de solicitação de endereço será aberta. Entre com o endereço **5000** e pressione **ENTER**. No campo de códigos será apresentado as correspondentes instruções da região de memória iniciada pelo endereço 5000.

Neste ponto, tanto o campo de ram externa, quanto o campo de códigos apontam para a mesma região de memória. Assim, se for alterado qualquer conteúdo no campo de ram externa, automaticamente será alterado o campo de códigos. A figura 6.14 apresenta a tela resultante desta operação.

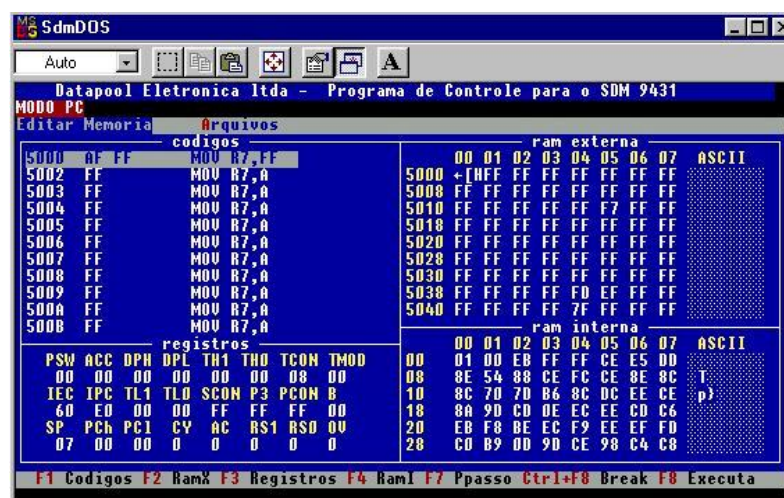



Figura 6.14 - Acesso ao campo de códigos.

Com as setas de movimentação poderá ser varrida a região de memória. Entretanto não poderá ser apresentado a linguagem mnemônica de endereços anteriores ao do topo da região destacada no campo de códigos. Isto poderá ser feito somente pela opção Verificar programa, do menu de editar memória, entrando com o endereço anterior desejado.

⇨ A tecla **ESC** retorna ao menu de comandos gerais.

⇨ Selecione a opção Registros no menu de Editar Memória.

Esta opção permite acesso ao campo de registros. O cursor estará posicionado neste campo. Isto possibilita alterar os conteúdos dos registros apresentados neste campo.

Através das setas de movimentação posicione o cursor no registro PCH e altere o seu conteúdo para 50H. Altere também o conteúdo de PCL para 00H mesmo que este valor já esteja escrito neste registro. O resultado está ilustrado na figura 6.15.

Figura 6.15 - Acesso ao campo de registros.

Ao ser modificado o conteúdo do registro PCL, automaticamente o campo de códigos será alterado para o endereço apontado pelo registro PC (PCH + PCL). Neste ponto o endereço apontado pelo PC estará destacado. Isto significa que o programa estará pronto para ser executado, tanto no modo direto (Opção F8), quanto no modo passo a passo (Opção F7). Isto será analisado posteriormente.

⇨ A tecla **ESC** retorna ao menu de comandos gerais.

⇨ Selecione a opção **Bits Ender** do menu de editar memória.

⇨ Será apresentada a tela da figura 6.16.

Figura 6.16 - Acesso aos bits endereçáveis.

➩ Uma janela de apresentação de "status" do bit endereçável será aberta.

Para acessar o bit correspondente entre com o endereço do mesmo. Por exemplo, entre com o endereço E7 e pressione **ENTER**. O correspondente conteúdo deste endereço será apresentado. Neste ponto o conteúdo poderá ser confirmado, ou alterado. Altere o valor para 1 e pressione **ENTER**.

O endereço E7 corresponde ao bit mais significativo do acumulador. Portanto, o conteúdo do acumulador estará apresentado o valor 80H. Os bits do acumulador são acessados pelos endereços de E0H até E7H, indo do bit menos significativo até o bit mais significativo. Repita a operação de acesso aos bits endereçáveis do acumulador e faça outras alterações.

➩ A alteração do bit retorna ao menu de comandos gerais.

➩ Selecione a opção **Ench Ram iNt.** do menu de editar memória.

➩ Será apresentada a tela da figura 6.17.

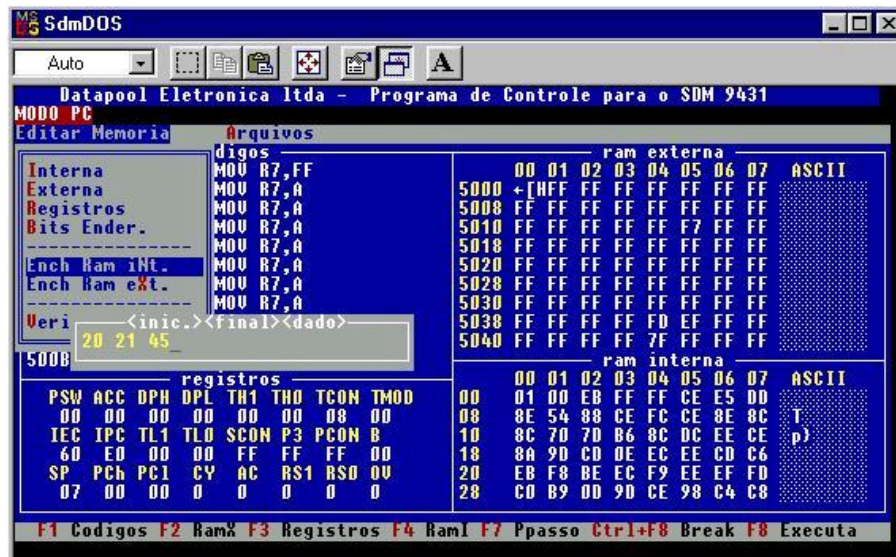



Figura 6.17 - Tela de preenchimento de área de Ram interna.

Uma janela de solicitação de endereços e do valor de preenchimento será aberta.

Deve ser colocados o endereço inicial, o endereço final e o valor que será escrito nesta região de memória interna. A área de memória interna do 8031 vai de 00H até 7FH. Por exemplo entre com o endereço inicial 28H, o endereço final 2FHm o dado 45H e pressione **ENTER**. A mensagem **"operação realizada"** será temporariamente apresentada e o sistema retorna ao menu de comandos gerais.

- ➪ Selecione a opção Ench Ram eXt. do menu de editar memória.
- ➪ Será apresentada a tela da figura 6.18.

Figura 6.18 - Tela de preenchimento de área de Ram externa.

Uma janela de solicitação de endereços e do valor de preenchimento será aberta.

Deve ser colocados o endereço inicial, o endereço final e o valor que será escrito nesta região de memória externa. A área de memória externa do 8031 vai de 0000H até FFFFH. No Módulo SDM 9431 a área disponível para o usuário vai de 5000H até BFFFH, quando estiver usando 32K bytes de memória, ou de 5000H até 5FFFH quando a opção for de 8K bytes de memória.

Por exemplo, entre com o endereço inicial 5040H, com o endereço final 5047H e com o dado AA e pressione **ENTER**. Entre os valores de endereços e dados deve existir um espaço em branco. A mensagem **"operação realizada"** será temporariamente apresentada e o sistema retorna ao menu de comandos gerais.

Analisados os comandos de edição de memória, retornaremos aos comandos de arquivos. Selecione novamente o menu de arquivos. Além da opção carregar, analisada anteriormente, existirá a opção **Salvar**, destinada a armazenagem de uma região de memória em unidades de disquete do computador PC. Selecione esta opção. Uma janela de solicitação de endereços e do arquivo.hex onde será armazenado os conteúdos será aberta, conforme apresentado na figura 6.19.

Figura 6.19 - Tela de armazenagem de arquivo.

Como exemplo entre com o endereço inicial **5000**, com o endereço final **501D**, com o caminho (drive, diretório, subdiretório, etc) e o nome **B:\contador.hex**. Uma mensagem de **"Aguarde"** será apresentada temporariamente enquanto os dados estiverem sendo transferidos e o sistema retornará ao menu de comandos gerais.

A última opção do menu de arquivos é a de Fim (DOS) que retorna o computador ao sistema operacional DOS. Isto poderá ser feita através de Ctrl + C, que também abortará o programa retornando ao DOS.

6.2.4 LINHA DE COMANDOS DE FUNÇÕES

A linha inferior da tela do programa de controle para o SDM 9431 contém os comandos de funções disponíveis para este sistema. Estes comandos farão um acesso mais rápido aos quadros dos campos de funções e também permitirão execuções de programas no modo passo a passo, no modo "breakpoint" e no modo direto.

O comando **F1 - Códigos** é usado para acessar ao campo de códigos. Pressionando-se a tecla F1, o cursor estará no campo de códigos e através das setas de movimentação poderá ser varrido este campo. A tecla **ESC** retorna ao menu de comandos gerais.

O comando **F2 - RamX** é usado para acessar ao campo de ram externa, permitindo verificações e alterações nos endereços em destaque. A tecla **ESC** retorna ao menu de comandos gerais.

O comando **F3 - Registros** é usado para acessar ao campo de registros, permitindo alterações nos registros em destaque. A tecla ESC retorna ao menu de comandos gerais.

O comando **F4 - RamI** é usado para acessar ao campo de ram interna, permitindo verificação ou alterações nos endereços em destaque. A tecla **ESC** retorna ao menu de comandos gerais.

O comando **F7 - Ppasso** é usado para a execução do programa no modo passo a passo, ou seja, somente uma instrução por vez. Para isto carregue o PC com o endereço inicial da execução. Por exemplo, carregue com o valor PCH = 50H e PCL = 00H (PC = 5000H) mesmo que este valor já esteja sendo apresentado, carregue novamente. Ao ser carregado o valor 00H no registro PCL o campo de códigos será atualizado e o endereço apontado por PC estará em destaque neste campo.

⇨ Neste ponto o programa poderá ser executado no modo passo a passo.

Pressione a tecla F7 uma vez. A primeira instrução foi executada. Esta instrução move para o endereço 81H, que é o endereço do registro SP, o valor imediato 2FH. Portanto, este valor aparecerá no registro SP do campo de registros e o PC apontará para o endereço 5003H.

Pressione a tecla F7 novamente. A segunda instrução foi executada. Esta instrução move o valor imediato 00H para o acumulador e o PC apontará para o endereço 5005H.

Pressione a tecla F7 novamente. A terceira instrução foi executada. Esta instrução é uma chamada para um subrotina interna do sistema cujo endereço inicial é 10AAH. Assim o PC será carregado com este endereço e o campo de códigos apresentará os correspondentes conteúdos desta região de memória.

Sucessivas operações da tecla F7 executarão as instruções subsequentes do programa em análise. Durante a operação passo a passo o usuário poderá verificar ou alterar qualquer área de memória, ou registros do processador, através dos menus e campos correspondentes.

O comando **F8 - Executa** é usado para fazer a execução direta do programa. Para isso, o contador de programa deverá estar apontando para a posição inicial do programa a ser executado. O programa Cont_Dec.Hex carregado anteriormente é um contador decimal que apresenta a sua contagem no display de cristal líquido e também na porta paralela P1.

Para verificar a sua operação, desligue o módulo. Usando oito fios conecte os bits da porta P1 (De P1.0 a P1.7 do barramento identificado por CP1) e localizado abaixo da memória RAM do sistema, com os correspondentes led's L0 a L7, localizados na parte inferior esquerda do Módulo SDM 9431.

- ⇨ Saia do programa SDM, retornando ao DOS.
- ⇨ Recarregue novamente o programa SDM.
- ⇨ Ligue o módulo.
- ⇨ Carregue o programa Cont_Dec novamente.

Carregue o PC com o valor 5000H. O campo de códigos apresentará o programa a ser executado.

Pressione a tecla **F8**, para a execução do programa. No display será apresentado a contagem decimal e nos led's aparecerão o valor BCD correspondente.

Como este programa está em loop o sistema de comunicação estará travado. É possível abortar o programa SDM e o contador continuará operando. Entretanto, para retornar ao controle do sistema pelo programa SDM é necessário que o Módulo SDM 9431 sofra um RESET e que haja um retorno ao DOS, para abrir novamente a comunicação entre o PC e o sistema SDM 9431.

O comando **Ctrl + F8** é usado para introduzir um "breakpoint", ou seja, um ponto de parada, a fim de executar depurações mais rápidas que no modo passo a passo.

IMPORTANTE

A execução do programa no modo "breakpoint" utiliza a interrupção externa INT1 para poder ser implementada. Assim, a interrupção INT1 não poderá ser utilizada, caso o programa seja executado neste modo.

Também o jump JP5, deverá estar posicionado para o lado esquerdo da conexão (entre os pinos identificados por GND e INT1).

Na utilização normal da interrupção externa INT1 este jump deverá estar posicionado para o lado direito da conexão (entre os pinos INT1 e INT1#).

Como exemplo de utilização, coloque o programa Cont_Dec.Hex no campo de códigos. Para isto utilize a opção **Verificar Prog**. no endereço 5000H.

Mova o cursor para o endereço **5010H** e pressione **Ctrl + F8**. Assim este endereço aparecerá em destaque no campo de código, indicando um endereço de parada.

- ⇨ Carregue o PC com 5000H.
- ⇨ Pressione a tecla F8.
- ⇨ O programa será executado até o endereço 5010H, parando neste ponto.

Portanto, permitirá a análise de endereços e registros afetados pelo programa até o ponto de parada. O usuário poderá continuar a executar o programa a partir deste endereço.

Por exemplo, poderá executar outro trecho usando "breakpoint". Para isto, pressione a tecla **ESC**. Entre no campo de códigos, pressionando a tecla **F1**. Mova o cursor para o endereço **5018H** e pressione **Ctrl + F8**. Novamente será destacado o ponto de parada.

Neste ponto o PC está carregado com o valor do último ponto de parada. Portanto, ao pressionar a tecla **F8** o programa será executado até o próximo ponto de parada.

Para executar instruções no modo passo a passo, a partir do último ponto de parada, entre no campo de códigos, pressionando **F1**, e pressione sucessivamente a tecla **F7**.

CAPÍTULO 7 - EXPERIÊNCIAS DE PROGRAMAÇÃO

Neste capítulo serão desenvolvidas experiências de programação com o ***Sistema SDM 9431***. Os programas apresentados utilizam as sub-rotinas internas usadas pelo programa monitor e que estão disponíveis para o usuário.

Detalhes sobre o uso destas sub-rotinas são apresentadas no Capítulo 4 - Sub-rotinas do Sistema SDM 9431. Os endereços destas sub-rotinas são apresentadas na tabela 7.1.

SUB-ROTINA	ENDEREÇO DE CHAMADA
AC_DSP	10E7h
AD	145Fh
ASCII	114Ch
CLS_DSP	10AAh
DA	1471h
DELAY	11C8h
DPT_DSP	1121h

SUB-ROTINA	ENDEREÇO DE CHAMADA
DSP_COM	109Ah
DSP_DAT	10FFh
LE_DADO	0F00h
LE_DAD1	0F27h
LE_TEC	1002h
MENS	110Fh
MONITOR	01C0h

Tabela 7.1 - Endereços das Sub-rotinas

Os microcontroladores da família 8051 dividem as regiões de memória em: memória de programa, onde residem as instruções, e em memória de dados, onde residem os dados. A memória de programa é uma memória somente de leitura.

Para possibilitar o desenvolvimento de programas, o ***Sistema SDM 9431*** possui a memória RAM acessada como memória de dados e como memória de programa.

O endereçamento utilizado para a memória está apresentado na tabela 7.2.

ENDEREÇO	DISPOSITIVO
0000h - 3FFFh	ROM
4000h - BFFFh	RAM
C000h - DFFFh	Livre
E000h - FFFFh	Dispositivos periféricos

Tabela 7.2 - Endereçamento da memória.

Dos 16kbytes disponíveis de ROM, aproximadamente 12kbytes foram ocupados pelo programa monitor e os 4kbytes restantes, de 3000h até 3FFFh, estão disponíveis para o usuário.

As posições iniciais da memória RAM são utilizadas pelo programa monitor. Assim, **o usuário deverá iniciar os seus programas a partir do endereço 5000h**. Quando for usada uma RAM de 64kbytes, o endereço final será BFFFh, quando for usada uma RAM de 8kbytes, o endereço final será 5FFFh.

Os endereços de C000h até DFFFh estão livres, para possibilitar o desenvolvimento de circuitos. O sistema possui esta faixa decodificada, através do sinal ***MS6***, disponíveis no barramento, para minimização de circuitos adicionais. Quando habilitado possui saída em nível baixo.

Os dispositivos periféricos externos do Sistema SDM 9431 fazem parte da região de endereçamento destinada a entradas ou saídas de dados, conforme apresentado na tabela 7.2.

Também neste caso, o sistema já possui endereços decodificados e disponíveis para minimização de circuitos. São os sinais ***I05***, ***I06*** e ***I07*** do conector ***CON12*** ou do barramento ***CP2***. As faixas de endereços para estes sinais, que quando habilitados possuem saída em nível baixo, são: de F400h até F7FFh para I05; de F800h até FBFFh para I06 e de FC00h até FFFFh para I07.

A seguir serão apresentados programas exemplos, destinados ao aprendizado das instruções do 8031 e da operação do Sistema SDM 9431.

São possíveis três maneiras de carregamento dos programas:

1. O usuário poderá carregar o programa por edição direta dos códigos operacionais (OPCODE), nos endereços correspondentes, usando a função INS/VER, quando o sistema operar no MODO TECLADO (Capítulo 2).
2. O usuário poderá carregar o programa por edição direta dos códigos operacionais (OPCODE), nos endereços correspondentes, usando a opção EXTERNA do menu de Editar Memória, quando o sistema estiver operando no MODO PC (Capítulo 3).
3. O usuário poderá editar o programa em linguagem Assembly (mnemônicos), converter o arquivo para o padrão HEX e carregar o arquivo convertido através da opção CARREGAR do menu de Arquivos, operando tanto no MODO PC, quanto no MODO TECLADO.

7.1 EXPERIÊNCIA 1: INICIALIZAÇÃO (RESET)

7.1.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.1.2 PROCEDIMENTO

- Se o sistema estiver selecionado para operar no modo Teclado, ligar o equipamento ou pressionar a tecla **RESET** no teclado ou no canto superior esquerdo do módulo, caso o mesmo já esteja ligado.
- Para usar o modo PC o sistema deve estar selecionado para operação no modo PC. Executar o software SDM-9431 e ligar o Módulo SDM-9431 para iniciar a comunicação ou pressionar a tecla **RESET**, se for solicitada a inicialização do sistema. Em ambos os casos, o sistema será inicializado.
- Operando no modo teclado, ou no modo PC, anotar o conteúdo dos seguintes registros:
-

REGISTRO	VALOR
PC	
ACC	
PSW	
SP	

REGISTRO	VALOR
B	
DPL	
DPH	
DPTR	

7.1.3 OBSERVAÇÕES

A tecla REG_ESP é usada para verificação de registros especiais, quando no modo teclado. Os seus conteúdos aparecem no campo Registros, quando no modo PC, para o entendimento destes registros leia o capítulo ***"Registros de Função Especial"*** no Manual de Teoria do Módulo SDM-9431.

- Os endereços dos registros especiais são (o registro DPTR é formado pelos registros DPH e DPL):

REGISTRO	B	DPL	DPH	SP
ENDEREÇO	F0h	82h	83h	81h

- Os valores encontrados correspondem aos valores de inicialização, efetuado pelo reset são apresentados na tabela seguinte.

REGISTRO	**VALOR**
PC	0000h
ACC	00h
B	00h
PSW	00h
SP	07h
DPTR	0000h
P0-P3	FFh
IP	XXX00000 B
IE	0XX00000 B
TMOD	00h
TCON	00h
THO	00h
TLO	00h
TH1	00h
TL1	00h
SCON	00h
SBUF	indeterm.
PCON (NMOS)	0XXXXXXX B
PCON (CMOS)	0XXX0000 B

- O registro PSW estará inicializado com o valor 00h. Isto significa que todos os seus bits estarão zerados. Portanto, os bits RS1 e RS0 estarão selecionando o banco de registros zero.

  O banco de registros zero contém os registros R0 até R7. Estes registros serão acessados respectivamente pelos endereços 00h até 07h, da RAM interna. Isto significa que o SP estará apontando para o registro R7.

7.2 EXPERIÊNCIA 2: MODOS DE ENDEREÇAMENTOS

Para melhor compreensão desta experiência deve ser lido o item ***Modos de Endereçamentos*** do capítulo ***Conjunto de Instruções da Família 8051***, do Manual de Teoria.

7.2.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.2.2 PROCEDIMENTO

- Carregar o programa seguinte, que visa familiarizar o usuário com os modos de endereçamento.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 28		mov A, # 28h	Acc ← 28h
5002	F9		mov R1, A	R1 ← Acc
5003	74 35		mov A, # 35h	Acc ← 35h
5005	90 50 18		mov DPTR, # 5018h	DPTR ← dado do end.
5008	F7		mov @R1, A	(R1) ← Acc
5009	F0		movx @ DPTR, A	(DPTR) ← Acc
500A	E5 81		mov A, 81h	Acc ← 81h
500C	93		movc A, @ A+DPTR	Acc ← (DPTR + Acc)
500D	12 01 C0		lcall MONITOR	

- A estrutura da instrução na representação mnemônica, ou em linguagem assembly, obedece a seguinte ordem: ***OPCODE DESTINO FONTE.*** Assim, a instrução ***MOV A, R0*** move o conteúdo de R0 para o acumulador.
- Alterar o conteúdo do registro ***SP*** para 2Fh.
- Executar o programa no modo passo a passo (ver Capítulo 5 deste manual), carregando o PC com 5000h, executando uma instrução por vez.

- Ao executar a ***primeira instrução***, o PC foi carregado com o valor _______h, que corresponde ao endereço da próxima instrução que será executada. A instrução executada carregou o acumulador com o valor _______h. *O modo de endereçamento desta instrução é o modo imediato. Isto significa que o byte seguinte ao opcode será transferido para o registro em questão (acumulador).*

- Executar o ***segundo passo*** do programa.

  A instrução executada move o conteúdo do acumulador para o registro _______. O modo de endereçamento desta instrução é o modo de endereçamento de registro, onde o registro utilizado pertence ao banco de registros selecionado. Como o registro PSW contém o valor 00h, o banco de registros 0 está selecionado. Assim os registros de R0 a R7 serão os correspondentes endereços 00h até 07h da Ram interna.

  Examinar o endereço 01h da Ram interna. O seu conteúdo será _______.

  Na execução desta instrução o PC está com o valor _______h. O acumulador com o valor _______h.

- Executar o ***terceiro passo***.

  A instrução executada é uma instrução imediata que move o valor _______h para o acumulador. O PC estará com o valor _______h.

- Executar o ***quarto passo***.

  A instrução executada é uma instrução imediata que move o valor _______h para o registro DPTR. Examinar o registro DPTR.

  O PC estará com o valor _______h.

- Executar o ***quinto passo***.

  A instrução MOV @R1, A é uma instrução no modo de endereçamento indireto, onde o registro R1 aponta para um endereço da Ram interna onde estará o operando. Neste caso R1 contém o valor _______h (veja E_REG, banco 0, registro 1 para modo teclado ou Ram interna, endereço 01h, para modo PC). *Portanto, a instrução move o conteúdo do acumulador para este endereço*.

  Examinar o endereço de Ram interna 01h, no modo PC, ou o registro 1, do banco 0, no modo teclado. O seu conteúdo será _______h.

  Neste ponto o PC está com o valor _______h.

  No modo de endereçamento indireto os registros R0 e R1, do banco de registros selecionados serão usados como apontadores de endereço do operando.

- Executar o ***sexto passo***.

  A instrução MOV @DPTR, A é uma instrução no modo de endereçamento indireto, na qual o registro DPTR aponta para um endereço da Ram externa onde está o

operando. Neste caso o registro DPTR contém o valor _______h. *Portanto a instrução move o conteúdo do acumulador para este endereço.*

Examine o endereço de Ram externa 5018h. O seu conteúdo será _______h.

- Executar o ***sétimo passo***.

  A instrução MOV A, 81h é uma instrução no modo de endereçamento direto. Isto significa que o byte seguinte ao opcode é um campo de endereço de oito bits. Portanto somente a Ram de dados interna (dos endereços de 00h até 7Fh) e os registros de função especial (dos endereços de 80h até FFh) é que poderão ser endereçados diretamente.

  O endereço 81h corresponde ao registro de função especial SP ("stack pointer"). Assim o acumulador foi carregado com o valor _______h. Examinar o conteúdo do registro SP. Este contém o valor _______h.

  O PC estará com o valor _______h.

- Executar o ***oitavo passo***.

  A instrução MOVC A, @A + DPTR é uma instrução no modo de endereçamento indexado, que é usado somente para leitura de memória de programa. Este modo se destina a leitura de tabelas armazenadas em uma área de programa. Estas são denominadas tabelas "Look-up". Por exemplo podem ser tabelas de conversões (senos, logarítmos), ou tabelas de mensagens. Assim, o conteúdo do endereço formado pela soma A + DPTR será transferido para o acumulador.

  No passo anterior o acumulador tinha o valor _______h e o DPTR está com o valor _______h. Portanto o conteúdo do endereço formado por (A + DPTR) = _______h foi transferido para o acumulador.

  Examinar a posição de endereço de Ram externa 501Fh. O seu conteúdo será _______h.

  O PC está com o valor _______h.

- Executar a ***última instrução***, LCALL 01C0, que é uma chamada de retorno ao programa monitor. *Esta deverá ser utilizada no final dos programas, para que, após a execução dos mesmos, o usuário retorne ao controle do sistema.*
- Pressionar a tecla reset, se estiver operando no modo teclado, ou pressionar a tecla ESC se operando no modo PC, para retornar ao programa monitor.

7.2.3 OBSERVAÇÕES

Deve-se notar que programas cíclicos (em loop) somente serão encerrados por interrupções ou pelos botões de reset do módulo.

7.3 EXPERIÊNCIA 3: INSTRUÇÕES ARITMÉTICAS

Para melhor compreensão desta experiência deve ser lido o item ***Instruções Aritméticas*** do capítulo ***Conjunto de Instruções da Família 8051***, do Manual de Teoria.

7.3.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.3.2 PROCEDIMENTO

- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções aritméticas.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 05		mov A, # 05h	Acc ← 05
5002	24 5A		add A, # 90	Acc + 90
5004	78 20		mov R0, # 20h	R0 ← 20
5006	F6		mov @ R0, A	(R0) ← Acc
5007	26		add A, @ R0	Acc + conteúdo de R0
5008	25 20		add A, 20h	Acc ← Acc + (20)
500A	74 05		mov A, # 05h	Acc ← 05
500C	34 07		addc A, # 07h	Acc ← Acc + 07 + carry
500E	74 05		mov A, # 05h	Acc ← Acc + 05
5010	04		inc A	Acc ← Acc + 1
5011	75 F0 07		mov B, # 07h	B ← 07
5014	A4		mul AB	multiplica BA
5015	74 23		mov A, # 23h	Acc ← 23
5017	7F 48		mov R7, # 48h	R7 ← 48
5019	2F		add A , R7	Acc ← Acc + dado de R7
501A	D4		da A	ajuste decimal do Acc
501B	12 01 C0		lcall MONITOR	

- Executar o programa no modo passo a passo, completando a tabela seguinte.

PASSO	PC =	REGISTROS E ENDEREÇOS VERIFICADOS
-	5000h	
1		Acc =
2		Acc = PSW =
3		R0 = Endereço 00h =
4		Endereço 20h =
5		Acc = PSW = R0 = Endereço 20 h =
6		Acc = PSW =
7		Acc = PSW = Carry =
8		Acc =
9		Acc =
10		Acc =
11		B = Endereço F0h =
12		A x B = (BA) = _______h
13		Acc =
14		R7 = Endereço 07h =
15		Acc =
16		Acc =

- Retornar ao programa monitor.

7.3.3 OBSERVAÇÕES

- No ***passo 1*** o acumulador foi carregado imediatamente com o número 05h.
- No ***passo 2*** a instrução add A, #90 irá somar ao acumulador o valor decimal 90, que corresponde ao valor hexadecimal 5A. Na nomenclatura mnemônica, os números em hexadecimal devem ser seguidos da letra "h", caso contrário serão interpretados como valores decimais e convertidos para o correspondente valor hexadecimal.

  A soma efetuada resulta em:

	0 5 H			0 0 0 0 0 1 0 1
+	5 A H	=	+	0 1 0 1 1 0 1 0
	5 F H			0 1 0 1 1 1 1 1

  O registro PSW terá o valor 00h, ou seja

CY	AC			0V			P
0	0	0	0	0	0	0	0

O resultado da soma obtida (5Fh) não gerou carry, não gerou carry auxiliar, não gerou overflow e o bit de paridade é par (0 = par).

- No ***passo 3*** o registro R0 é carregado imediatamente com o valor 20h, ou seja R0 é o endereço 00h da Ram interna.
- No ***passo 4*** o valor do acumulador é transferido indiretamente para a posição de memória apontada por R0, ou seja, endereço 20h.
- No ***passo 5*** é realizada a soma indireta do conteúdo do endereço apontado por R0 e do conteúdo do acumulador, ou seja, Acc = 5Fh e (R0) = (20h) = 5Fh, resultado em:

		5 F H			0 1 0 1 1 1 1 1
Acc + (R0) =	+	5 F H	=	+	0 1 0 1 1 1 1 1
		B E H			1 0 1 1 1 1 1 0

O registro PSW terá o valor 44h, ou seja:

CY	AC				0V		P
0	1	0	0	0	1	0	0

O resultado da soma obtida BEh não gerou carry, gerou carry auxiliar, gerou overflow e o bit de paridade é par.

- No ***passo 6*** é realizada a soma direta do conteúdo do endereço 20h e do conteúdo do acumulador, ou seja, Acc = BEh e (20h) = 5Fh, resultando em:

		B E H			1 0 1 1 1 1 1 0
Acc + (20H) =	+	5 F H	=	+	0 1 0 1 1 1 1 1
		1 D H		1 (CY)	0 0 0 1 1 1 0 1

O registro PSW terá o valor C0h, ou seja:

CY	AC				0V		P
1	1	0	0	0	0	0	0

O resultado da soma gerou um carry, gerou um carry auxiliar, não gerou overflow e o bit de paridade é par.

- No ***passo 7*** é carregado imediatamente o valor 05h no acumulador.

O valor do registro PSW é mantido em C0h. Isto significa que as instruções de transferência de dados não afetam os "flags" de condições.

O carry continua com o valor 1.

- No ***passo 8*** é realizada uma soma com carry e com o valor imediato 07h, resultando em:

			0 5 H			0 0 0 0 0 1 0 1
			0 7 H			0 0 0 0 0 1 1 1
Acc + 07H + CY	=	+	1	=	+	1
			0 D H			0 0 0 0 1 1 0 1

- No ***passo 9*** é carregado o valor imediato 05h no acumulador.
- No ***passo 10*** o conteúdo do acumulador é incrementado.
- No ***passo 11*** o registro B, cujo endereço é F0h, é carregado com o valor 07h.
- No ***passo 12*** é realizada a multiplicação dos conteúdos dos registros A e B. O resultado de uma multiplicação de dois números de oito bits é um número de 16 bits. Assim o par de registros BA é usado como saída da instrução multiplicação. Portanto,

A x B = (BA), ou seja, 06h x 07h = 002Ah

que corresponde ao valor 42 decimal.

- No ***passo 13*** é carregado imediatamente o valor 23h no acumulador.
- No ***passo 14*** é carregado imediatamente o valor 48h no registro R7, ou seja, endereço 07h da Ram interna.
- No ***passo 15*** é realizada a soma dos conteúdos do acumulador e do registro R7, resultando em:

			2 3 H			0 0 1 0 0 0 1 1
Acc + R7	=	+	4 8 H	=	+	0 1 0 0 1 0 0 0
			6 B H			0 1 1 0 1 0 1 1

- No ***passo 16*** é realizado o ajuste aritmético decimal através da instrução da A. O ajuste aritmético decimal é realizado em operações de soma de números BCD e irá somar o número 6 toda vez que o resultado da soma ultrapassar o número 9, ou seja:

Acc	=	6 B			0 1 1 0 1 0 1 1
	+	0 6	=	+	0 0 0 0 0 1 1 0
		7 1			0 1 1 1 0 0 0 1

7.4 EXPERIÊNCIA 4: INSTRUÇÕES LÓGICAS

7.4.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431 | Desktop ou Notebook

7.4.2 PROCEDIMENTO

- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções lógicas.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 37		mov A, # 37h	Acc ← 37
5002	54 0F		anl A, # 0Fh	'E' imediato entre 0F e Acc
5004	23		rl A	desloca Acc a esquerda
5005	79 20		mov R1, # 20h	R1 ← 20
5007	F7		mov @ R1, A	(R1) ← Acc
5008	E4		clr A	Acc = 0
5009	74 25		mov A, # 25h	Acc ← 25
500B	47		orl A, @ R1	'OU' entre Acc e (R1)
500C	F5 26		mov 26h, A	End 26 ← Acc
500E	63 26 FF		xrl 26h, 0FFh	'XOR' entre End (26) e FF
5010	C4		swap A	troca bit's do Acc
5011	13		rrc A	desloca Acc para direita
5012	13		rrc A	com carry
5013	12 01 C0		lcall MONITOR	

- Executar o programa no modo passo a passo, completando a tabela seguinte.

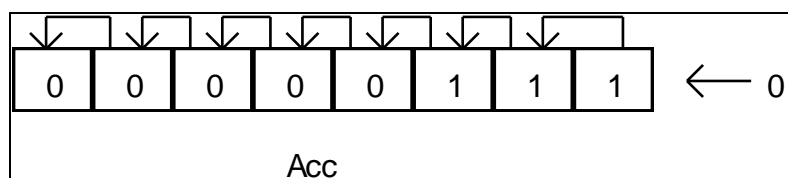
PASSO	PC =	REGISTROS E ENDEREÇOS VERIFICADOS
-	5000h	
1		Acc =
2		Acc =
3		Acc =
4		R1 = Endereço 01h =
5		Endereço 20h =
6		Acc =
7		Acc =
8		Acc =
9		Endereço 26h = Acc =
10		Endereço 26h = Acc =
11		Acc = CY =
12		Acc = CY =
13		Acc = CY =

- Retornar ao programa monitor.

7.4.3 OBSERVAÇÕES

- No ***passo 1*** o acumulador foi carregado com o valor imediato 37h.
- No ***passo 2*** foi realizada a lógica E (AND) entre o conteúdo do acumulador e a máscara 0Fh resultando em:

Acc →	3 7 H			0 0 1 1 0 1 1 1
(E)	0 F H	=	(E)	0 0 0 0 1 1 1 1
	0 7 H			0 0 0 0 0 1 1 1

- No ***passo 3*** o conteúdo do acumulador sofre um deslocamento lógico para a esquerda, resultando em 0Eh, ou seja:

- No ***passo 4*** o registro R1 foi carregado imediatamente com 20h.
- No ***passo 5*** o conteúdo do acumulador foi movido indiretamente para o endereço apontado por R1, ou seja, endereço 20h.
- No ***passo 6*** o conteúdo do acumulador foi zerado.
- No ***passo 7*** o acumulador foi carregado imediatamente com o valor 25h.

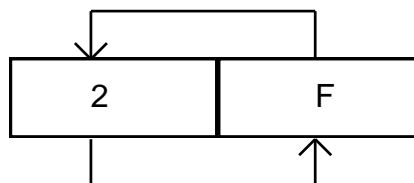
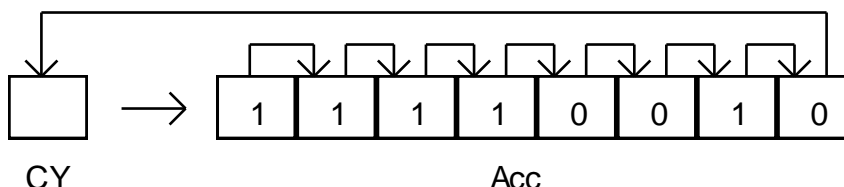
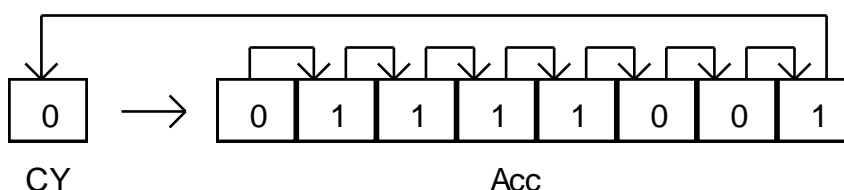
- No ***passo 8*** foi realizada indiretamente a lógica OU entre o conteúdo do acumulador e do endereço apontado pelo registro R1, ou seja, endereço 20h, resultando em:

Acc ⟶	2 5 H			0 0 1 0 0 1 0 1
(OU)	0 E H	=	(OU)	0 0 0 0 1 1 1 1
	2 F H			0 0 1 0 1 1 1 1

- No ***passo 9*** o acumulador foi transferido diretamente para o endereço 26h.
- No ***passo 10*** foi realizada a lógica OU exclusiva entre o conteúdo do endereço 26 e o valor imediato FFh. Como resultado, os bits do endereço 26h foram invertidos, ou seja,

Acc ⟶	2 F			0 0 1 0 0 1 0 1
(XOR)	F F	=	(XOR)	1 1 1 1 1 1 1 1
	D 0			1 1 0 1 0 0 0 0

Esta operação foi realizada sem que algum dos valores envolvidos passassem pelo acumulador. Note que o conteúdo do acumulador foi preservado. Este tipo de instrução é chamada de leitura - modificação - escrita, onde o conteúdo do endereço é lido, alterado e o resultado é escrito no mesmo endereço. Será útil para o manuseio de bits de uma porta paralela.

- No ***passo 11*** foi realizado uma troca entre o "nibble" (meio byte) mais significativo e o "nibble" menos significativo do acumulador, ou seja

- No ***passo 12*** foi realizado uma rotação para a direita, com carry, ou seja:

- No ***passo 13*** foi realizada uma nova rotação para a direita, com carry, resultando no valor 3Ch do conteúdo do acumulador e carry igual a 1, ou seja:

7.5 EXPERIÊNCIA 5: INSTRUÇÕES DE TRANSFERÊNCIA DE DADOS

Para melhor compreensão desta experiência leia o item ***Instruções de Transferência de Dados*** do capítulo ***Conjunto de Instruções da Família 8051***, do Manual de Teoria.

Nas experiências 2, 3 e 4 já foram executadas algumas instruções de transferência de dados. Este grupo está subdividido em instruções de dados usando a Ram interna e instruções de transferência de dados usando a memória de dados externa.

Os registros R0 e R1 são usados como apontadores de endereços para a Ram interna.

O registro DPTR é usado como apontador de endereço para a Ram externa.

7.5.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.5.2 PROCEDIMENTO

- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções complementares de transferência de dados.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 20		mov sp, # 20h	inicializa stack pointer
5003	74 26		mov A, # 26h	Acc ← 26
5005	79 18		mov R1, # 18h	R1 ← 18
5007	F7		mov @ R1, A	mover Acc indireto
5008	90 50 38		mov DPTR, # 5038h	mover imediato p/ DPTR
500B	F0		movx @ DPTR, A	mover Acc indireto
500C	C0 01		push 1	pilha ← R1
500E	C9		xch A, R1	trocar Acc e R1
500F	D7		xchd A, @ R1	trocar nibble de Acc - R1
5010	D0 01		pop 1	R1 ← pilha
5012	12 01 C0		lcall MONITOR	

Executar o programa no modo passo a passo, completando a tabela seguinte.

PASSO	PC =	REGISTROS E ENDEREÇOS VERIFICADOS
-	5000h	
1		SP = Endereço 81h =
2		Acc =
3		R1 = Endereço 01h =
4		Endereço 18h =
5		DPH = Endereço 83h = DPL = Endereço 82h = DPTR =
6		Endereço 5038h =
7		SP = Endereço 81h = Endereço 21h = _________ (= R1) R1 = Acc =
8		R1 = Acc = Endereço 26 h =
9		Acc = Endereço 26h = SP = Endereço 81h =
10		SP = Endereço 81h = R1 =

- Retornar ao programa monitor.

7.5.3 OBSERVAÇÕES

- No ***passo 1*** o registro SP foi carregado imediatamente com 20h.
- No ***passo 2*** o acumulador foi carregado imediatamente com 26h.
- No ***passo 3*** o registro R1 foi carregado imediatamente com 18h.
- No ***passo 4*** o conteúdo do acumulador foi transferido indiretamente para o endereço apontado por R1, ou seja, o endereço 18h.
- No ***passo 5*** o registro DPTR foi carregado imediatamente com o valor 5038h.
- No ***passo 6*** o conteúdo do acumulador foi transferido indiretamente para o endereço 5038h.
- No ***passo 7*** o conteúdo do registro R1 foi transferido para a pilha (stack). Note que a pilha reside na Ram interna e varia em sentido crescente. Assim a instrução PUSH primeiro incrementa o registro SP para depois copiar o dado. Esta instrução usa somente o modo de endereçamento direto.
- No ***passo 8*** a instrução XCH A, R1 efetuou a troca entre os dados do acumulador e do registro R1.

- No ***passo 9*** a instrução XCHD.A, @R1 efetuou a troca somente dos quatro bits inferiores do acumulador e do endereço apontado pelo registro R1, ou seja endereço 26h.
- No ***passo 10*** o registro R1 é carregado com o valor do topo da pilha (stack) e posteriormente, o registro SP é decrementado.

7.6 EXPERIÊNCIA 6: INSTRUÇÕES BOOLEANAS

Para melhor entendimento desta experiência deve ser lido o item ***Instruções Booleanas*** do capítulo ***Conjunto de Instruções da Família 8051***, do Manual de Teoria.

Esta família de processadores possui um conjunto completo de operações de bit, ou operações Booleanas.

Os endereços de bytes da Ram interna de 20h até 2Fh são também bits endereçáveis e os bits responderão aos endereços de bit de 00h até 7Fh, variando em ordem crescente desde o bit menos significativo do byte até o bit mais significativo do mesmo. A tabela seguinte mostra este endereçamento.

BYTE	ENDEREÇOS DO BITS	
	DE (LSB)	ATÉ (MSB)
20	00h	07h
21	08h	0Fh
22	10h	17h
23	18h	1Fh
24	20h	27h
25	28h	2Fh
26	30h	37h
27	38h	3Fh
28	40h	47h
29	48h	4Fh
2A	50h	57h
2B	58h	5Fh
2C	60h	67h
2D	68h	6Fh
2E	70h	77h
2F	78h	7Fh

Os registros de função especial com endereçamento terminado por 0h ou por 8h são também bits endereçáveis e respondem aos endereços de bits desde 80h até FFh. A tabela seguinte mostra o endereçamento de bits destes registros.

NOME DO REGISTRO	ENDEREÇO DE BYTE	ENDEREÇO DE BIT	
		DE (LSB)	ATÉ (MSB)
P0	80h	80h	87h
TCON	88h	88h	8Fh
P1	90h	90h	97h
SCON	98h	98h	9Fh
P2	A0h	A0h	A7h
IE	A8h	A8h	AFh
P3	B0h	B0h	B7h
IP	B8h	B8h	BFh
PSW	D0h	D0h	D7h
ACC	E0h	E0h	E7h
B	F0h	F0h	F7h

Note que pelo sistema utilizar RAM externa, o usuário não poderá, em seu programa, alterar os bits P3.6 e P3.7, respectivamente WR e RD da porta P3. Como as portas P0 e P2 estão sendo usadas para endereçamentos externos, os seus bits também não poderão ser acessados.

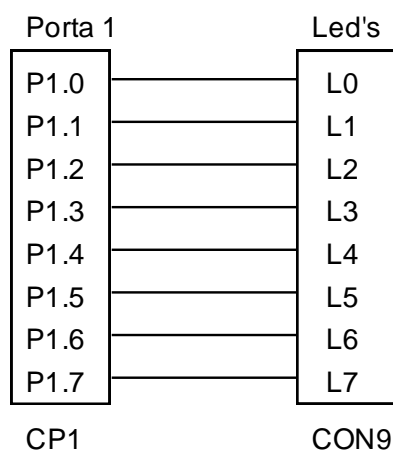
7.6.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.6.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1	Led's
P1.0	L0
P1.1	L1
P1.2	L2
P1.3	L3
P1.4	L4
P1.5	L5
P1.6	L6
P1.7	L7
CP1	CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções Booleanas.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 90 01		mov P1, # 01h	P1 ← 01
5003	75 20 D4		mov 20h, # 0D4h	End 20 ← D4
5006	A2 00		mov C, 00h	CY ← 00
5008	72 01		orl C, 01h	'OU' entre CY e 01
500A	92 90		mov 90h, C	End 90 ← CY
500C	A2 02		mov C, 02h	CY ← 02
500E	72 03		orl C, 03h	'OU' entre C e 03
5010	92 90		mov 90h, C	End 90 ← CY
5012	A2 04		mov C, 04h	CY ← 04
5014	B0 05		anl C, /05h	'E' entre CY e /05
5016	92 91		mov 91h, C	End 91 ← CY
5018	D3		setb C	seta o CY
5019	92 92		mov 92h, C	End 92 ← CY
501B	A2 06		mov C, 06h	CY ← 06
501D	30 07 01		jnb 07h, FIM	jump se bit 07 = 0
5020	B3		cpl C	inverte o estado do CY
5021	92 92	FIM:	mov 92h, C	End 92 ← CY
5023	12 01 C0		lcall MONITOR	

- Executar o programa no modo passo a passo, completando a tabela seguinte.

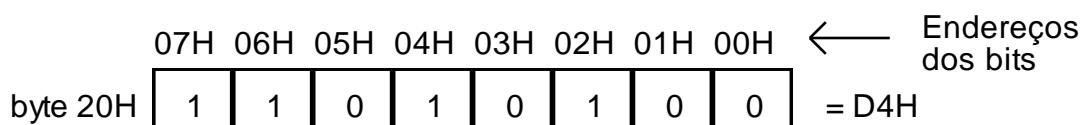
PASSO	PC =	REGISTROS E ENDEREÇOS VERIFICADOS
-	5000h	
1		
2		Endereço 20h =
3		CY =
4		CY =
5		P1 =
6		CY =
7		CY =
8		P1 =
9		CY =
10		CY =
11		P1 =
12		CY =
13		P1 =
14		CY =
15		
16		CY =
17		P1 =

- Retornar ao programa monitor.
- Não desconectar os fios da porta P1, caso vá realizar a Experiência 7, que utilizarão as mesmas ligações.

7.6.3 OBSERVAÇÕES

O bit de carry opera como acumulador para as instruções Booleanas. Este é o bit mais significativo do registro PSW.

- No ***passo 1*** o valor 01h foi enviado para o endereço 90h, que é o endereço da porta 1, portanto este valor é apresentado nos leds.
- No ***passo 2*** o valor D4h é carregado no endereço de 20h. Este endereço é também bit endereçável. Os bits endereçáveis responderão aos endereços, conforme o esquema seguinte:

	07H	06H	05H	04H	03H	02H	01H	00H	← Endereços dos bits
byte 20H	1	1	0	1	0	1	0	0	= D4H

- No ***passo 3*** o bit de endereço 00h é movido para o carry.
- No ***passo 4*** é realizada uma operação lógica **OU** entre CY e o bit de endereço 01h, portanto,

CY = 0 **OU** *0 = 0.*

- No ***passo 5*** o valor de CY é armazenado no bit de endereço 90h, que é o bit 0 da porta P1, portanto o led L0 apagará.
- No ***passo 6***, CY é carregado com o valor do bit de endereço 02h.
- No ***passo 7*** é realizada uma operação lógica **OU** entre CY e o bit de endereço 03h, portanto,

CY = 1 **OU** *0 = 1.*

- No ***passo 8*** o valor de CY é armazenado no bit de endereço 90h, que é o bit 0 de P1, portanto L0 acenderá.
- No ***passo 9*** CY é carregado com o bit de endereço 04h.
- No ***passo 10*** é realizada uma operação lógica **E** entre CY e o complemento do bit de endereço 05h, portanto,

CY = 1 **E** *0 = 1.*

- No ***passo 11*** CY é armazenado no bit de endereço 91h, que é o bit 2 de P1, portanto o led L2 acenderá.
- No ***passo 12*** CY é levado para 1.
- No ***passo 13*** CY é armazenado no bit de endereço 92h, que é o bit 3 de P1, portanto o led L3 acenderá.
- No ***passo 14*** CY é carregado com o valor do bit de endereço 06h.
- No ***passo 15*** a instrução JNB 07, FIM irá efetuar o salto se o bit do endereço 07h for zero. Caso contrário, continua normalmente.
- No ***passo 16***, como a instrução JNB não foi realizada, será efetuado o complemento de CY.
- No ***passo 17***, o valor de CY será armazenado no bit de endereço 92h, apagando o led L3.
- O conjunto de instruções dos passos 14, 15 e 16 realiza a operação lógica ***OU-exclusivo*** entre os valores dos bits de endereços 06h e 07h, ou seja:

MOV C, bit 1

JNB bit 2, salto

CPL C

salto: (neste ponto C = bit 1 ⊕ bit 2)

Se o bit 2 for zero o valor do bit 1 será mantido.

Se o bit 2 for um o valor do bit 1 será complementado.

7.7 EXPERIÊNCIA 7: INSTRUÇÕES DE DESVIO

Para melhor entendimento desta experiência deve ser lido o item ***Instruções de Desvio*** do capítulo ***Conjunto de Instruções da Família 8051***, do Manual de Teoria.

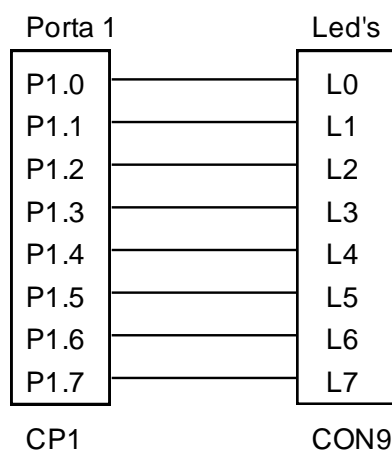
7.7.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.7.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1	Led's
P1.0	L0
P1.1	L1
P1.2	L2
P1.3	L3
P1.4	L4
P1.5	L5
P1.6	L6
P1.7	L7
CP1	CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções de desvio.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 00		mov A, 00h	Acc ← 00
5002	F5 90	REPETE:	mov P1, A	P1 ← Acc
5004	75 00 05		mov 0, # 05h	R0 ← 05
5007	75 01 FF	SALTO3:	mov 1, # 0FFh	R1 ← FF
500A	75 02 FF	SALTO2:	mov 2, # 0FFh	R2← FF
500D	DA FE	SALTO1:	djnz R2, SALTO1	decrementa e salta
500F	D9 F9		djnz R1, SALTO2	se Reg ≠ 0
5011	D8 F4		djnz R0, SALTO3	
5013	04		inc A	Acc + 1
5014	B4 81 EB		cjne A,#81h,REPETE	desvia se Acc ≠ 81
5017	12 01 C0		lcall MONITOR	

- Execute o programa no modo direto.
- O programa irá contar em binário, apresentando os valores nos leds. A contagem inicial em 00h e irá parar no número 80h, retornando ao monitor.
- Não desconectar os fios da porta P1, caso vá realizar a Experiência 8, que utilizarão as mesmas ligações.

7.7.3 OBSERVAÇÕES

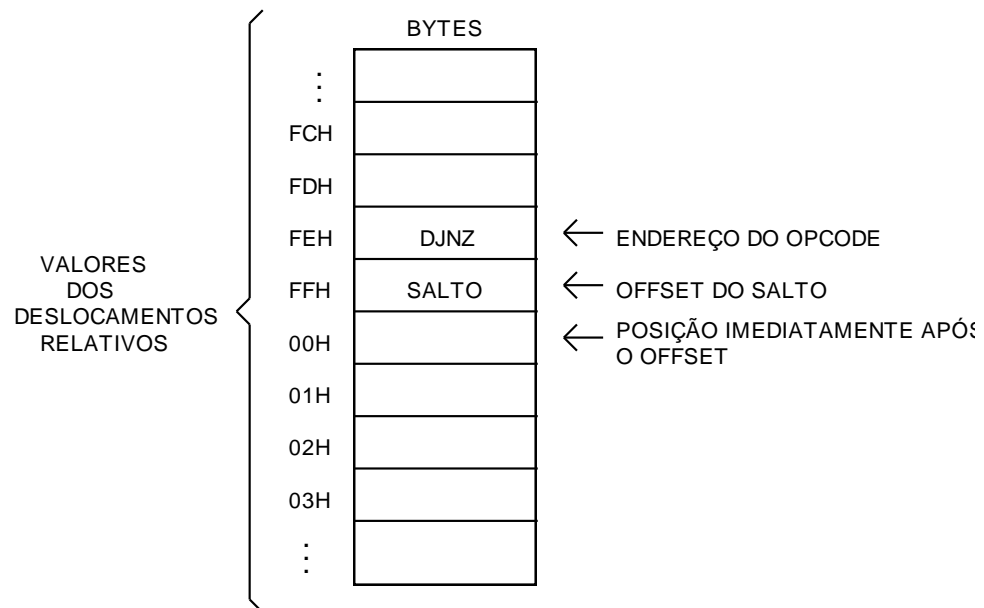
Um Loop de atraso usando os registros R1, R2 e R3 foi implementado, para a visualização da contagem nos led's. Estes trechos do programa poderá ser transformado em uma subrotina de atraso.

As instruções de desvio utilizadas neste programa foram:

- ***DJNZ Rn, salto*** → Esta instrução decrementa o conteúdo do registro Rn e efetua um salto se o valor do conteúdo de Rn não for zero. O tamanho do salto é especificado através de deslocamento relativo, ou seja, saltos sinalizados.

  Por exemplo os códigos **DA FE** correspondem a instrução **djnz R2, FE**. No programa o opcode, **DA**, está colocado no endereço **500Dh** e o deslocamento (offset), **FEh**, está no endereço **500Eh**. Assim, quando o microcontrolador buscar o opcode, **DA**, o PC será incrementado e conterá o valor **500Eh**. Ao identificar a instrução **djnz**, o microcontrolador inicia o ciclo de busca do deslocamento e o PC será incrementado e conterá o valor **500Fh**. O deslocamento **FEh** é um valor negativo e vale menos dois (-2). Portanto, ao executar a instrução, o valor -2 será somado ao PC, que terá o valor **500Fh**, e será efetuado um salto para o endereço **500Dh**.

  Isto mostra que o valor do deslocamento (offset) é somado relativamente (valores sinalizados) ao conteúdo do PC para obter o endereço final do salto. Como regra

geral, o endereço imediatamente após o offset equivale ao endereço de salto zero. Endereços anteriores terão valores negativos de saltos, endereços posteriores terão valores positivos de saltos. A figura seguinte esquematiza estes valores.

- ***CJNE A, #81h, salto*** → Esta instrução compara o conteúdo do acumulador com o valor imediato 81h e efetua o salto se os valores nã***o*** forem iguais. Também neste caso, o valor do deslocamento será um número relativo (sinalizado). A instrução é composta pelo opcode seguido do byte imediato de comparação, seguido do deslocamento relativo. No programa o opcode desta instrução está no endereço 5014h, o byte de comparação está no endereço 5015h e o deslocamento relativo está no endereço 5016h. Isto significa que após a busca total desta instrução o PC estará apontando para o endereço 5017h. Deseja-se um salto para o endereço 5002h, ou seja, retornar 21 posições. Portanto o offset será de $-21_{(10)}$ = EBh. O valor é sempre representado em complemento de dois, ou seja:

$$21_{(10)} = 0001\ 0101$$

Logo: -21 = 1110 1010 + 1 = 1110 1011 = EBh

Outras instruções do grupo de instruções de desvio serão apresentadas em experiências posteriores.

7.7.4 PROBLEMA PROPOSTO

- Através de edição direta na Ram externa, alterar o programa, para que o mesmo conte até 40h e até 20h, apresentando estes valores nos leds.

7.8 EXPERIÊNCIA 8: USO DE SUBROTINAS

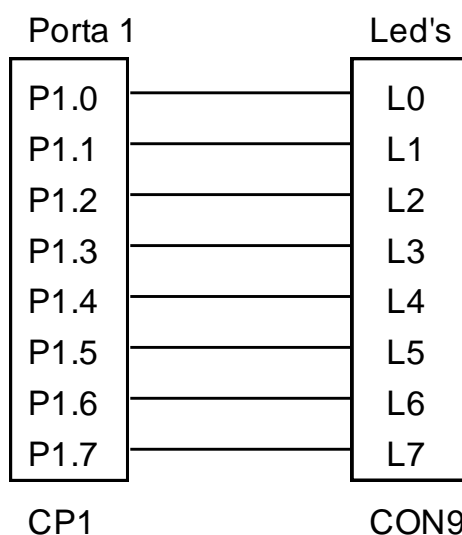
7.8.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.8.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1	Led's
P1.0	L0
P1.1	L1
P1.2	L2
P1.3	L3
P1.4	L4
P1.5	L5
P1.6	L6
P1.7	L7
CP1	CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que tem a mesma função do programa da Experiência 7, ou seja, apresenta uma contagem na porta P1 (entretanto, o atraso entre as apresentações dos valores é feito através de uma subrotina de atraso).

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2Fh	inicializa stack pointer
5003	74 00		mov A, # 00h	Acc ← 00
5005	F5 90	REPETE:	mov P1, A	P1 ← Acc
5007	12 50 11		lcall ATRASO	busca subrotina tempo
500A	04		inc A	Acc ← Acc + 1
500B	B4 81 F7		cjne A,#81h,REPETE	jump se Acc ≠ 81
500E	12 01 C0		lcall MONITOR	
5011	C0 00	ATRASO:	push 0	pilha ← R0
5013	C0 01		push 1	pilha ← R1
5015	C0 02		push 2	pilha ← R2
5017	78 05		mov R0, # 05h	R0 ← 05
5019	79 FF	SALTO3:	mov R1, # 0FFh	R1 ← FF
501B	7A FF	SALTO2:	mov R2, # 0FFh	R2 ← FF
501D	DA FE	SALTO1:	djnz R2, SALTO1	decrementa e salta
501F	D9 F9		djnz R1, SALTO2	se reg ≠ 0
5021	D8 F4		djnz R0, SALTO3	
5023	D0 02		pop 2	R2 ← pilha
5025	D0 01		pop 1	R1 ← pilha
5027	D0 00		pop 0	R0 ← pilha
5029	22		ret	retorne da subrotina

- Execute o programa no modo direto.
- Nos leds serão apresentados os valores hexadecimais da contagem em questão. O programa será encerrado, retornando ao monitor, quando, nos leds, aparecer o valor 80h.
- Não desconectar os fios da porta P1, caso vá realizar a Experiência 9, que utilizará as mesmas ligações.

7.9 EXPERIÊNCIA 9: USO DO DISPLAY

Para melhor entendimento desta experiência devem ser lidos o item ***Display de Cristal Líquido*** e o item ***Rotinas para o Display*** do Manual de Experiências.

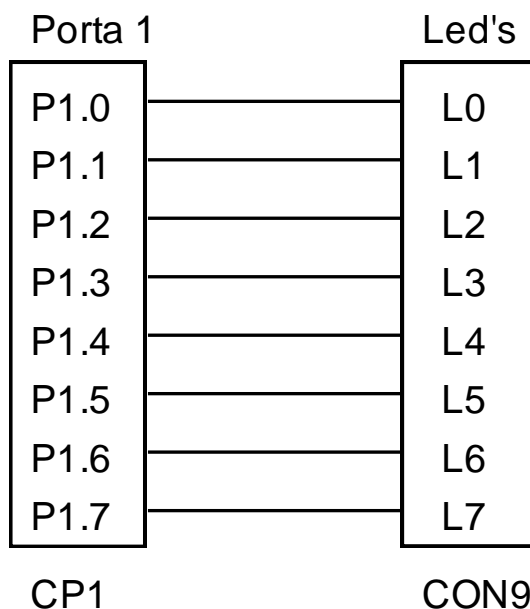
7.9.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.9.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1		Led's
P1.0	—	L0
P1.1	—	L1
P1.2	—	L2
P1.3	—	L3
P1.4	—	L4
P1.5	—	L5
P1.6	—	L6
P1.7	—	L7
CP1		CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que visa apresentar rotinas de utilização do display.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, #2Fh	inicializa o stack pointer
5003	74 00		mov A, #00h	Acc ← 00
5005	12 10 AA	REPETE:	lcall CLR_DSP	limpa display
5008	F5 90		mov P1, A	P1 ← Acc
500A	12 10 E7		lcall ac_dsp	display ← Acc
500D	04		inc A	Acc ← Acc + 1
500E	12 50 13		lcall ATRASO	busca subrotina ATRASO
5011	80 F2		sjmp REPETE	
5013	C0 00	ATRASO:	push 0	pilha ← R0
5015	C0 01		push 1	pilha ← R1
5017	C0 02		push 2	pilha ← R2
5019	78 05		mov R0, #05h	R0 ← 05
501B	79 FF	SALTO3:	mov R1, #0FFh	R1 ← FF
501D	7A FF	SALTO2:	mov R2, #0FFh	R2 ← FF
501F	DA FE	SALTO1:	djnz R2, SALTO1	decr. R2 e desvia se ≠ 0
5021	D9 FA		djnz R1, SALTO2	decr. R1 e desvia se ≠ 0
5023	D8 F6		djnz R0, SALTO3	decr. R0 e desvia se ≠ 0
5025	D0 02		pop 2	R2 ← pilha
5027	D0 01		pop 1	R1 ← pilha
5029	D0 00		pop 0	R0 ← pilha
502B	22		ret	retorne da subrotina

- O programa irá apresentar a contagem hexadecimal no display e na porta P1.
- Para encerrar a execução do programa pressione a tecla RESET.

7.9.3 OBSERVAÇÕES

- O acionamento da tecla reset do módulo irá interromper a comunicação entre o módulo e o microcomputador PC. Assim, o módulo deve ser reconectado ao microcomputador.
- O reset não altera os conteúdos da memória Ram e, portanto, o programa continua instalado a partir do endereço 5000h.

7.9.4 PROBLEMA PROPOSTO

- Alterar o programa da Experiência 9, para que a contagem seja feita em decimal no display e apresentada em BCD na porta P1.

7.10 EXPERIÊNCIA 10: MENSAGENS NO DISPLAY

Para o melhor entendimento da experiência leia os itens ***Display de Cristal Líquido*** e ***Rotinas para o Display*** do Manual de Experiências.

7.10.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.10.2 PROCEDIMENTO

- Carregar o programa seguinte, que escreve mensagens no display.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2Fh	carregar stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	90 50 27		mov dptr,#men1	DPTR ← sub rotina MEN1
5009	12 11 0F		lcall MENS	busca de sub rotina MENS
500C	74 C0		mov a ,# 0C0h	Acc ←C0h(comando de linha)
500E	12 10 9A		lcall DSP_COM	comando do display
5011	90 50 19		mov dptr,#men2	DPTR ← sub rotina MEN2
5014	12 11 0F		lcall MENS	busca de sub rotina MENS
5017	80 FE	REPETE:	sjmp REPETE	
5019	0D 20 20	MEN2 :	db 13,	
501C	20 45 4C			
501F	45 54 52			***" ELETRONICA "***
5022	4F 4E 49			
5025	43 41			
5027	0C 20 20	MEN1 :	db 12,	
502A	20 20 44			
502D	41 54 41			***" DATAPOOL "***
5030	50 4F 4F			
5033	4C			

- Executar o programa no modo direto.
- Para encerrar a execução do programa deve ser pressionada a tecla reset do módulo.

7.10.3 OBSERVAÇÕES

- O acionamento da tecla reset do módulo irá interromper a comunicação entre o módulo e o microcomputador PC. Assim, o módulo deve ser reconectado ao microcomputador.
- O reset não altera os conteúdos da memória Ram e, portanto, o programa continua instalado a partir do endereço 5000h.
- A subrotina ***CLR_DSP*** limpa o display e posiciona o cursor na primeira coluna da primeira linha.
- A subrotina ***MENS*** escreve uma mensagem no display. Para isto o primeiro byte da mensagem deverá conter o número de caracteres da mensagem. Por exemplo, a mensagem men1 terá 13 caracteres sendo 03 espaços em branco e 10 letras. Os caracteres deverão ser escritos no código ASCII, conforme a tabela 2 do item ***"Display de Cristal Líquido"*** do Manual de Experiências.
- Através do comando C0h a subrotina ***DSP_COM*** posiciona o cursor na primeira coluna da segunda linha.

7.10.4 PROBLEMA PROPOSTO

- Alterar a mensagem para que o nome do usuário apareça no display.

7.11 EXPERIÊNCIA 11: COMANDOS PARA O DISPLAY

Para o melhor entendimento da experiência leia os itens ***Display de Cristal Líquido*** e ***Rotinas para o Display*** do Manual de Experiências.

7.11.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.11.2 PROCEDIMENTO

- Carregar o programa seguinte, que efetuará um deslocamento de mensagem no display.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2Fh	carregar stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	74 80	REPETE:	mov a, # 80h	Acc ← 80h
5008	12 10 9A		lcall DSP_COM	comando do display
500B	90 50 34		mov dptr, #MENG	DPTR ← sub rotina MENG
500E	12 11 0F		lcall MENS	busca de sub rotina MENS
5011	74 18		mov a, #18h	Acc ← 18h
5013	12 10 9A		lcall DSP_COM	comando do display
5016	12 50 1B		lcall ATRASO	busca sub rotina de atraso
5019	80 EB		sjmp REPETE	
501B	C0 00	ATRASO:	push 0	pilha ← R0
501D	C0 01		push 1	pilha ← R1
501F	C0 02		push 2	pilha ← R2
5021	78 02		mov R0 , # 02h	R0 ← 02h
5023	79 FF	SALTO3:	mov R1, # 0FFh	R1 ← FFh
5025	7A FF	SALTO2:	mov R2, # 0FFh	R2 ← FFh
5027	DA FE	SALTO1:	djnz R2, SALTO1	
5029	D9 FA		djnz R1, SALTO2	decr. jump se registro ≠ 0
502B	D8 F6		djnz R0, SALTO3	
502D	D0 02		pop 2	R2 ← pilha
502F	D0 01		pop 1	R1 ← pilha
5031	D0 00		pop 0	R0 ← pilha
5033	22		ret	retorne da sub rotina
5034	19 20 44	MENG:	db 19,	

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5037	41 54 41			
503A	50 4F 4F			
503D	4C 20 45			***" DATAPOOL***
5040	4C 45 54			***ELETRONICA***
5043	52 4F 4E			***LTDA "***
5046	49 43 41			
5049	20 4C 54			
504C	44 41 2E			

- Executar o programa no modo direto.
- Para encerrar a execução do programa deve ser pressionada a tecla reset do módulo.

7.11.3 OBSERVAÇÕES

- O acionamento da tecla reset do módulo irá interromper a comunicação entre o módulo e o microcomputador PC. Assim, o módulo deve ser reconectado ao microcomputador.
- O reset não altera os conteúdos da memória Ram e, portanto, o programa continua instalado a partir do endereço 5000h.
- A subrotina ***DSP_COM*** é usada para enviar um comando para o display conforme a tabela de comandos do display de cristal líquido e também posicionar o cursor no display através do envio de um comando com o código equivalente ao endereço do caracter. Por exemplo, o comando C6, enviado pelo subrotina DSP_COM, posicionará o cursor na sétima coluna da segunda linha.

7.11.4 PROBLEMA PROPOSTO

- Elaborar um programa que escreva o primeiro nome do usuário a partir da sexta coluna da primeira linha.

7.12 EXPERIÊNCIA 12: LEITURA DE TECLADO

Para o melhor entendimento da experiência devem ser lidos os itens ***Teclado*** e ***Subrotinas para o Teclado,*** do Manual de Experiências.

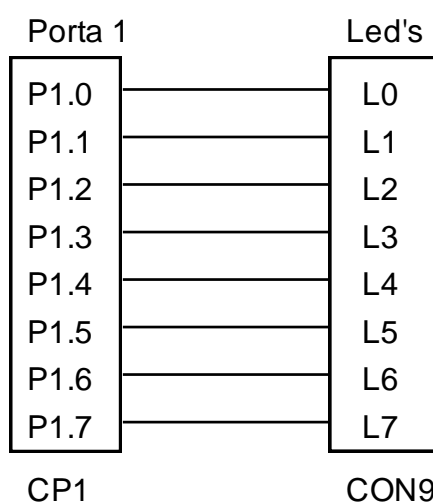
7.12.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.12.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1	Led's
P1.0	L0
P1.1	L1
P1.2	L2
P1.3	L3
P1.4	L4
P1.5	L5
P1.6	L6
P1.7	L7
CP1	CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que visa apresentar rotinas de utilização do teclado.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, #2Fh	inicializa o stack pointer
5003	75 90 00		mov P1, #00h	P1 ← 00
5006	74 00		mov A, #00h	Acc ← 00
5008	12 10 AA		lcall CLR_DSP	limpa display
500B	12 10 02	REPETE:	lcall LE_TEC	espera tecla pressionada
500E	33		rlc a	desloca Acc a direita
500F	40 0B		jc FIM	
5011	13		rrc a	desloca Acc a esquerda
5012	12 10 AA		lcall CLR_DSP	limpa display
5015	F5 90		mov P1, A	P1 ← Acc
5017	12 10 E7		lcall AC_DSP	display ← Acc
501A	80 EF		sjmp REPETE	
501C	12 01 C0	FIM:	lcall monitor	

- A tecla hexadecimal pressionada, de 0 a F, terá o seu valor apresentado no display e na porta P1.
- Quaisquer outras teclas pressionadas irão abortar o programa.

7.13 EXPERIÊNCIA 13: USO DO CONVERSOR A/D

Para melhor entendimento desta experiência devem ser lidos os itens ***O Conversor Análogo-Digital*** e ***Subrotinas de Uso Geral*** do Manual de Experiências.

7.13.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

Voltímetro (escala de 10Vcc) ou multímetro

7.13.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre o sinal ***P4*** e a entrada ***EA1***, que corresponde a entrada zero do conversor análogo-digital.
- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que exemplifica o uso do conversor A/D

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2F	inicializa o stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	74 02	REPETE:	mov A, # 02h	Acc ← 02
5008	12 10 9A		lcall DSP_COM	comando do display
500B	74 0C		mov A, # 0Ch	Acc ← 0C
500D	12 10 9A		lcall DSP_COM	comando do display
5010	90 E0 00		mov DPTR,# EA1	DPTR ← EA1
5013	12 14 5F		lcall AD	busca de sub rotina AD
5016	12 10 E7		lcall AC_DSP	display ← Acc
5019	80 EB		sjmp REPETE	

- Executar o programa no modo direto.
- Conectar um voltímetro na entrada EA1, para verificar a tensão analógica.
- O programa irá converter o valor da tensão na entrada analógica, apresentando o seu correspondente valor digital no display, ou seja, o sistema opera como um voltímetro digital.
- Com uma chave de ajuste especial (como as mostradas na figura seguinte ou pequena chave de fenda) alterar a posição do trimpot P4, ajustando-o para obter uma leitura desejada no multímetro.
- Com o trimpot na posição zero volts, o display deve mostrar 00.
- Com o trimpot na posição 5 volts, o display deve mostrar FF.
- Para encerrar a execução do programa deve ser pressionada a tecla Reset.

ATENÇÂO!

Os trimpots exigem manuseio delicado e o uso de ferramentas adequadas. A aplicação de grandes esforços ou o acionamento muito rápido pode provocar danos ao componente.

7.13.3 OBSERVAÇÕES

- A faixa de tensão de 0 a +5V terá a variação digital de 00 até FF. Portanto, o degrau de variação medido pelo conversor será dada por:

  +5 volts / 256 = 0,01956 ≅ 0,02 volts/bit

  ou seja, uma variação de 0,02 volts corresponde a uma variação de um dígito binário. Assim, se o display apresentar o valor 14, tem-se:

  14h = 20(10) ⇒ 20 x 0,02 ≅ 0,4 volts

  Se o display apresentar 7F, tem-se:

  7Fh = 128(10) ⇒ 128 x 0,02 ≅ 2,5 volts

7.14 EXPERIÊNCIA 14: USO DO CONVERSOR D/A

Para melhor entendimento desta experiência devem ser lidos os itens ***O Conversor Digital-Analógico*** e ***Subrotinas de Uso Geral*** do Manual de Experiências.

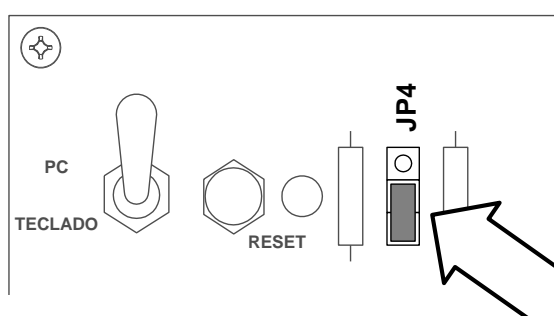
7.14.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

Voltímetro (escala de 10Vcc) ou multímetro

7.14.2 PROCEDIMENTO

- Com o equipamento desligado, colocar o jump J4 na posição indicada abaixo, que seleciona a faixa de o a +5V para a saída analógica.

- Conectar um voltímetro à saída DAC, no barramento superior do módulo.
- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que exemplifica o uso do conversor D/A

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2F	inicializa o stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	74 02	REPETE:	mov A, # 02h	Acc ← 02
5008	12 10 9A		lcall DSP_COM	comando do display
500B	12 0F 27		lcall LE_DAD1	lê duas teclas do teclado
500E	40 05		jc fim	se **ESC,** final de programa
5010	12 14 71		lcall DA	busca de sub rotina DA
5013	80 F1		sjmp REPETE	
5015	12 01 C0	fim:	lcall monitor	

- Executar o programa no modo direto.
- Neste programa o valor digital, em hexadecimal, introduzido no teclado será convertido no valor analógico correspondente.
- Após digitar o valor deve-se pressionar a tecla ENTER para a confirmação do mesmo.
- Para sair do programa pressione a tecla ESC do teclado do Módulo SDM 9431.

7.14.3 OBSERVAÇÕES

- O valor 00h será convertido para 0 volts.
- O valor FFh será convertido para +5 volts.
- Assim, quando selecionada a faixa de 0 a +5V, o degrau de conversão será de:

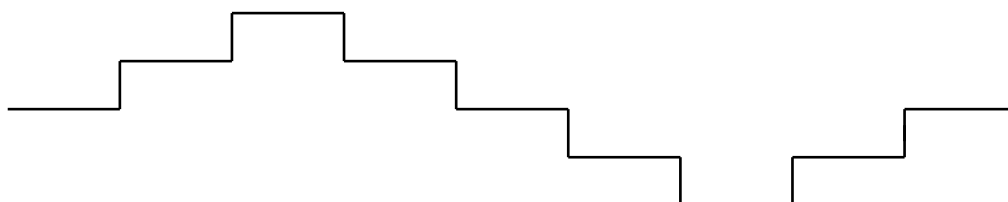
  $+12 \text{ volts} / 256 = 0{,}01956 \cong 0{,}2 \text{ volts/bit}$

  Portanto, digitando-se o valor 7F tem-se:

  $7Fh = 128_{(10)} \Rightarrow 128 \times 0{,}02 \cong 2{,}5 \text{ volts}$

- O jump JP4 seleciona a operação do DA na faixa de 0 a +5 volts (posicionado para baixo), ou na faixa -5 volts a +5 volts.

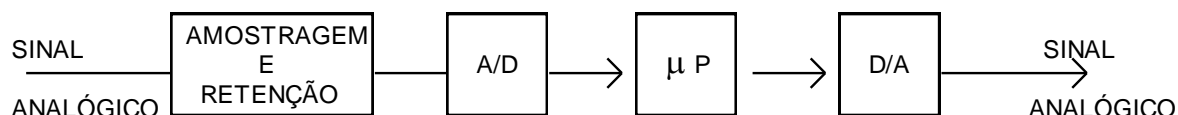
7.14.4 PROBLEMA PROPOSTO

- Escrever um programa que gere uma onda no formato seguinte, com intervalos de tempo constante.

7.15 EXPERIÊNCIA 15: SIMULAÇÃO DE UM CONTROLE REALIMENTADO DE SISTEMA

Na natureza os fenômenos possuem comportamentos contínuos no tempo e em sua amplitude. Ou seja, são informações analógicas. Assim antes do desenvolvimento dos sistemas digitais estes fenômenos, ou processos de controle dos mesmos eram implementados de maneira analógica.

Dificuldades eram encontradas na precisão destes controles. Por exemplo, supondo que no cálculo de um filtro analógico fosse necessário o uso de um capacitor de valor 0,012053 µF, a dificuldade de se obter um componente com tal precisão implicaria realizar um controle com uma certa variação na resposta do filtro.

Com o desenvolvimento dos sistemas digitais, os sinais analógicos puderam ser quantizados e digitalizados e o processo de controle poderá ser realizado através de programação. As variáveis digitais processadas poderão ser novamente convertidas em sinais analógicos e realimentadas no sistema.

A figura seguinte esquematiza o controle de um processo através de um sistema digital.

Como o processo de controle é implementado por programação, pode-se atribuir alto grau de precisão para as variáveis do processo. O sistema torna-se flexível e preciso com a única restrição feita pelo tempo de resposta do microprocessador.

Nesta experiência será adquirido um sinal analógico, efetuado um atraso no tempo e retornado este sinal, através do conversor digital-analógico. Isto pode representar o efeito de "eco". No lugar do atraso no tempo poderia ser implementado qualquer outro processamento, por exemplo, uma filtragem digital, uma transformada de Fourier-FFT, etc.

7.15.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

2 Voltímetro (escala de 10Vcc) ou 2 multímetro

7.15.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre o sinal ***P4*** e a entrada ***EA1***, que corresponde a entrada zero do conversor análogo-digital.
- Os sinais de entrada e saída devem ser verificados com o uso de dois voltímetros, ou de um osciloscópio, ligados aos pontos EA1 e DAC do Módulo SDM 9431.
- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que simula um controle de sistema

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2F	inicializa o stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	74 02	REPETE:	mov A, # 02h	Acc ← 02
5008	12 10 9A		lcall DSP_COM	comando do display
500B	74 C0		mov A, # 0C0h	Acc ← C0
500D	12 10 9A		lcall DSP_COM	comando do display
5010	90 E0 00		mov dptr, # EA1	DPTR ← EA1
5013	12 14 5F		lcall AD	busca sub rotina AD
5016	12 10 E7		lcall AC_DSP	mostra Acc no display
5019	12 11 C8	DELAY:	lcall DELAY	busca sub rotina de tempo
501C	12 14 71		lcall DA	busca sub rotina DA
501F	80 E5		sjmp REPETE	

- Executar o programa no modo direto.
- O programa irá converter o valor da tensão na entrada analógica, apresentando o seu correspondente valor digital no display e devolvendo o sinal, digitalizado e com retardo, através do conversor D/A.
- Com uma chave de ajuste especial (como as mostradas na figura seguinte ou pequena chave de fenda) alterar a posição do trimpot P4, ajustando-o para obter uma leitura desejada no multímetro conectado à entrada.
- Com o trimpot na posição zero volts, o display deve mostrar 00 e a saída do conversor D/A fornecer zero volts..
- Com o trimpot na posição 5 volts, o display deve mostrar FF e a saída do conversor D/A fornecer zero volts.
- Para encerrar a execução do programa deve ser pressionada a tecla Reset.

ATENÇÂO!

Os trimpots exigem manuseio delicado e o uso de ferramentas adequadas. A aplicação de grandes esforços ou o acionamento muito rápido pode provocar danos ao componente.

7.15.3 OBSERVAÇÕES

- O valor analógico da entrada EA1 será convertido para digital, apresentado no display, no formato hexadecimal e retornado para o conversor D/A.
- Os dois multímetros deverão apresentar as mesmas leituras. Porém, com o uso de um osciloscópio pode ser observado um pequeno atraso entre os sinais. Este retardo poderá ser alterado, conforme a subrotina de atraso utilizada.

7.16 EXPERIÊNCIA 16: RELÓGIO IMPLEMENTADO POR PROGRAMAÇÃO

7.16.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431 | Desktop ou Notebook

7.16.2 PROCEDIMENTO

- Carregar o programa seguinte, que transforma o ***Módulo SDM 9431*** em um relógio digital.

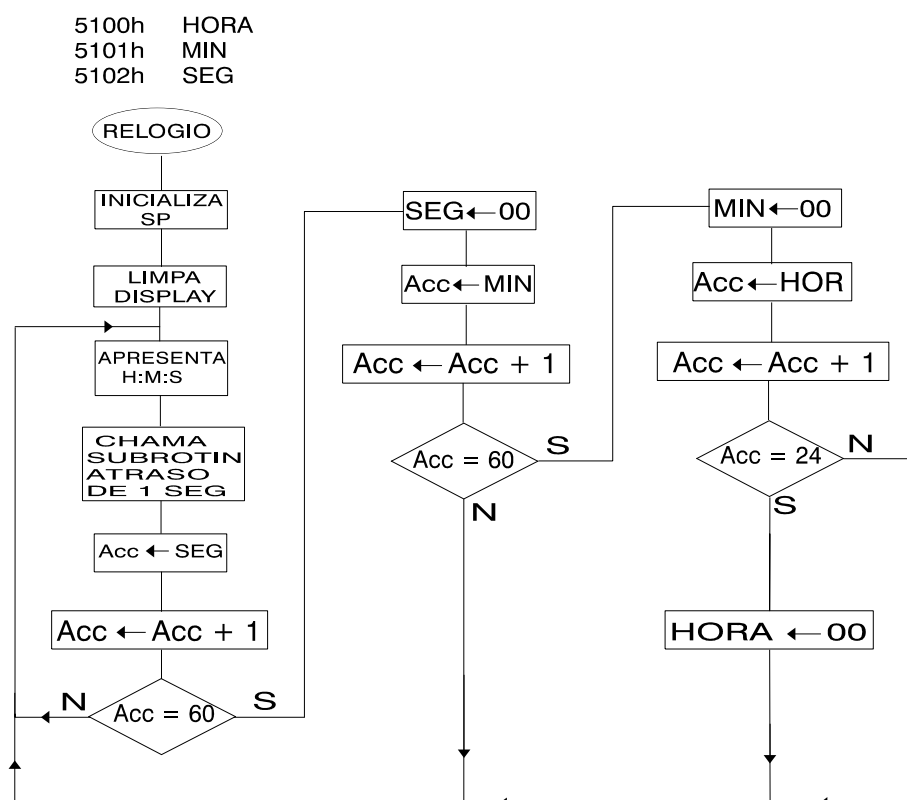
END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2Fh	inicializa stack pointer
5003	12 10 AA		lcall CLR_DSP	limpa display
5006	12 50 38	REPETE:	lcall MT_DSP	display ← hora
5009	12 50 7E		lcall UM_SEG	busca subrotina de 1 seg.
500C	90 51 02		mov DPTR, # 5102	DPTR ← **segundos**
500F	E0		movx A, @ DPTR	Acc ← (DPTR)
5010	24 01		add A , # 01h	Acc + 1
5012	D4		da A	ajuste decimal em Acc
5013	F0		movx @ DPTR, A	(DPTR) ← Acc
5014	B4 60 EF		cjne A,#60h,REPETE	incr. Acc e jump se ≠ 60
5017	74 00		mov A, # 00h	Acc ← 00
5019	F0		movx @ DPTR, A	(DPTR) ← Acc
501A	90 51 01		mov DPTR, # 5101	DPTR ← **minutos**
501D	E0		movx A, @ DPTR	Acc ← (DPTR)
501E	24 01		add A , # 01h	Acc ← Acc + 1
5020	D4		da A	ajuste decimal em Acc
5021	F0		movx @ DPTR, A	(DPTR) ← Acc
5022	B4 60 E1		cjne A,#60h,REPETE	incr. Acc e jump se ≠ 60
5025	74 00		mov A, # 00h	Acc ← 00
5027	F0		movx @ DPTR, A	DPTR ← Acc

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5028	90 51 00		mov DPTR, # 5100	DPTR ← **horas**
502B	E0		movx A, @ DPTR	Acc ← (DPTR)
502C	24 01		add A , # 01h	Acc ← Acc + 1
502E	D4		da A	ajuste decimal em Acc
502F	F0		movx @ DPTR, A	(DPTR) ← Acc
5030	B4 24 D3		cjne A,#24h,REPETE	incr. Acc e jump se ≠ 24
5033	74 00		mov A, # 00h	Acc ← 00
5035	F0		movx @ DPTR, A	(DPTR) ← Acc
5036	80 CE		sjmp REPETE	
5038	74 C0	MT_DSP	mov A, # 0C0h	Acc ← C0
503A	12 10 9A		lcall DSP_COM	comando do display
503D	90 51 00		mov DPTR, # 5100	DPTR ← **horas**
5040	E0		movx A, @ DPTR	Acc ← (DPTR)
5041	C4		swap A	trocas de bit's
5042	12 50 8B		lcall ASC_AL	
5045	12 10 FF		lcall DSP_DAT	display ← hora
5048	E0		movx A, @ DPTR	Acc ← (DPTR)
5049	12 50 8B		lcall ASC_AL	
504C	12 10 FF		lcall DSP_DAT	display ← hora
504F	74 3A		mov a, # ':'	Acc ← ': '
5051	12 10 FF		lcall DSP_COM	display ← ': '
5054	90 51 01		mov DPTR, # 5101	DPTR ← **minutos**
5057	E0		movx A, @ DPTR	Acc ← (DPTR)
5058	C4		swap A	trocas de bit's
5059	12 50 8B		lcall ASC_AL	
505C	12 10 FF		lcall DSP_DAT	display ← hora
505F	E0		movx A, @ DPTR	Acc ← (DPTR)
5060	12 50 8B		lcall ASC_AL	
5063	12 10 FF		lcall DSP_DAT	display ← hora
5066	74 3A		mov a, # ':'	Acc ← ': '
5068	12 10 FF		lcall DSP_COM	display ← ': '
506B	90 51 02		mov DPTR, # 5102	DPTR ← **segundos**
506E	E0		movx A, @ DPTR	Acc ← (DPTR)
506F	C4		swap A	trocas de bit's
5070	12 50 8B		lcall ASC_AL	
5073	12 10 FF		lcall DSP_DAT	display ← hora
5076	E0		movx A, @ DPTR	Acc ← (DPTR)
5077	12 50 8B		lcall ASC_AL	
507A	12 10 FF		lcall DSP_DAT	display ← hora
507D	22		ret	retorne da subrotina
507E	78 08	UM_SEG:	mov R0, # 08h	R0 ← 08

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5080	79 FF	um_seg0:	mov R1, # 0FFh	R1 ← FF
5082	7A F1	um_seg1	mov R2, # 0F1h	R2 ← 0F
5084	DA FE	um_seg2:	djnz R2, um_seg2	
5086	D9 FA		djnz R1, um_seg1	decr. reg. e jump se ≠ 0
5088	D8 F6		djnz R0, um_seg0	
508A	22		ret	retorne da subrotina
508B	54 0F	ASC_AL:	anl A, # 0Fh	'E' entre Acc e 0F
508D	24 30		add A, # 30h	Acc + 30h
508F	22		ret	retorne da subrotina

- Colocar o valor da hora, minuto e segundo nos respectivos endereços 5100h, 5101h e 5102h.
- Executar o programa no modo direto.

7.16.3 OBSERVAÇÕES

- O fluxograma correspondente a este programa é:

7.17 EXPERIÊNCIA 17: INTERRUPÇÃO

Para o melhor entendimento desta experiência devem ser lidos capítulo ***Interrupções*** e o texto sobre o registro TCON no capítulo ***Periféricos Internos do 8051***, ambos no Manual de Teoria do Módulo SDM 9431.

7.17.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431 Desktop ou Notebook

7.17.2 PROCEDIMENTO

- Carregar o programa seguinte, que mostra o funcionamento de uma interrupção.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
4230	02 50 2F		ljmp intr	desvia para endereço 502D
5000	75 81 2F		mov sp, # 2Fh	carregar stack pointer
5003	75 A8 81		mov ie, # 81h	reg. IE ← 81h
5006	75 88 01		mov tcon, #01h	reg. TCON ← 01
5009	74 00		mov a, # 0h	Acc ← 0h
500B	12 10 AA		lcall CLR_DSP	limpa display
500E	B4 01 FD	CONT:	cjne a,#01,cont	jump p/ CONT se Acc ≠ 01
5011	75 A8 00		mov ie, # 00h	reg. IE ← 00h
5014	90 50 23		mov dptr, #men	DPTR ← sub rotina MEN
5017	12 11 0F		lcall MENS	busca de sub rotina MENS
501A	12 10 02		lcall LE_TEC	espera tecla pressionada
501D	12 10 AA		lcall CLR_DSP	limpa display
5020	12 01 C0		lcall MONITOR	programa monitor
5023	0B 69 6E	MEN :	db 11,	
5026	74 65 72			***" INTERRUPÇÃO "***
5029	72 75 70			
502C	63 61 6F			
502F	75 A8 00	INTR:	mov ie, # 00h	reg. IE ← 00h
5032	74 01		mov a, #01h	Acc ← 01h

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5034	75 A8 81		mov ie, # 81h	reg. IE ← 81h
5037	32		reti	retorne da interrupção

- Executar o programa no modo direto.
- Pressionar a tecla INTER, que solicitará uma interrupção do sistema, que fará o programa em execução parar, atender à solicitação de interrupção e retornar à execução do programa principal.
- Pressionar uma tecla para encerrar a execução do programa principal.

7.17.3 OBSERVAÇÕES

- A tecla **INTR** está conectada à entrada **INT0** do microcontrolador 8031. Se algum circuito que faça uma solicitação de interrupção estiver conectado ao ponto INT0 do barramento CP1, esta tecla não deverá ser pressionada, pois a solicitação de interrupção deverá ser feita pelo circuito conectado à **INT0**.
- O processo de atendimento de interrupção da família 8051 é feito por endereços vetoriais, ou seja, quando a solicitação de interrupção **INT0** ocorrer, o processador irá para o endereço vetorial **0003h**, que está na ROM.
- Neste endereço da ROM foi colocada a instrução **LJMP 4230** que é um endereço da RAM. Porém, esta área de RAM é utilizada pelo programa monitor e não comportará uma rotina de serviço de interrupção muito grande. Portanto, no endereço 4230h deve-se colocar uma instrução **LJMP** para o endereço onde se deseja iniciar a rotina de serviço de interrupção INT0, o que foi feito neste programa.
- Deve-se utilizar, ou inicializar, os registros IP, IE e TCON, conforme a necessidade do manuseio de interrupções. Estes registros são inicializados com 00h pela tecla **RESET** do módulo SDM-9431.
- No Módulo SDM-9431 os endereços vetoriais da RAM que são acessados pelas interrupção são os listados na tabela seguinte. Assim, ao utilizar uma destas interrupções, no endereço correspondente deve-se colocar um **LJMP** para a posição inicial da rotina de serviço de interrupção.

FONTE DE INTERRUPÇÃO	ENDEREÇO VETORIAL DA RAM
IEO	4230h
TFO	4240h
IE1	4250h
TF1	4260h
RI + TI	4270h

7.18 EXPERIÊNCIA 18: USO DO TEMPORIZADOR INTERNO

Para melhor entendimento da experiência, leia o item ***Temporizadores / Contadores*** do capítulo ***Periféricos Internos do 8051*** e o capítulo ***Interrupção***, ambos do Manual de Teoria.

7.18.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.18.2 PROCEDIMENTO

- Carregar o programa seguinte, que apresenta os registros de controle para uso dos temporizadores internos do 8031.

Vetor para interrupção do temporizador 0

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
4240	02 50 24		jmp INTR	vetor p/ rot. de interrupção

Programa principal

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	75 81 2F		mov sp, # 2F	carrega stack pointer
5003	75 89 01		mov tmod, # 01h	tmod ← 01
5006	75 88 00		mov tcon, # 00h	tcon ← 00
5009	75 B8 00		mov ip, # 00h	ip ← 00
500C	75 A8 82		mov ie, # 82h	ie ← 82
500F	75 8C FF		mov tho, # 0FFh	tho ← FF
5012	75 8A FF		mov tl0, # 0FFh	tl0 ← FF
5015	12 10 AA		lcall CLR_DSP	limpa display
5018	74 0C		mov a, # 0Ch	comando do display
501A	12 10 9A		lcall DSP_COM	
501D	74 00		mov a, # 00h	Acc ← 00

501F	75 88 10		mov tcon, # 10h	tcon ← 10 iniciar temp.
5022	80 FE	REPETE:	sjmp REPETE	

Rotina para serviço de interrupção

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5024	75 88 00	INTR:	mov tcon, # 00h	tcon ← 00 parar temp.
5027	C0 E0		push Acc	sp ← Acc
5029	74 02		mov a, # 02h	comando do display
502B	12 10 9A		lcall DSP_COM	
502E	D0 E0		pop Acc	Acc ← sp
5030	12 10 E7		lcall AC_DSP	display ← Acc
5033	04		inc a	Acc + 1
5034	75 88 10		mov tcon, # 10h	tcon ← 10 iniciar temp.
5037	32		reti	retorne da interrupção

- Executar o programa no modo direto.

7.18.3 OBSERVAÇÕES

- O programa usa interrupções do temporizador/contador zero interno para efetuar uma contagem hexadecimal no display.
- A instrução MOV tmod, #01h prepara o temporizador/contador 0 para operar como temporizador. Isto significa que a geração de pulsos para o temporizador será baseada no próprio circuito oscilador do microcontrolador.
- A instrução MOV tcon, #00h deixa o temporizador zero inoperante.
- A instrução MOV ip, #00h programa todas as interrupções para operarem no nível zero de prioridade.
- A instrução MOV ie, #082h habilita a operação da interrupção gerada pelo temporizador zero.
- O temporizador 0 é carregado com o valor FFFFh e efetua uma contagem decrescente. Quando o mesmo chegar no valor 0000h haverá uma solicitação de interrupção.
- A instrução MOV tcon, #10h liga a operação do temporizador.
- Neste ponto o programa entrada em loop.
- Quando ocorrer a interrupção, o programa é vetorado para o endereço 4240h e neste endereço efetua um salto para a rotina de serviço da interrupção do temporizador 0.

- Nesta rotina o valor do acumulador é apresentado no display e incrementado, retornando ao ponto de parada do programa principal.
- Assim, sucessivas interrupções farão com que uma contagem hexadecimal seja apresentada no display.

7.19 EXPERIÊNCIA 19: DESLOCAMENTO DE BITS NA PORTA P1

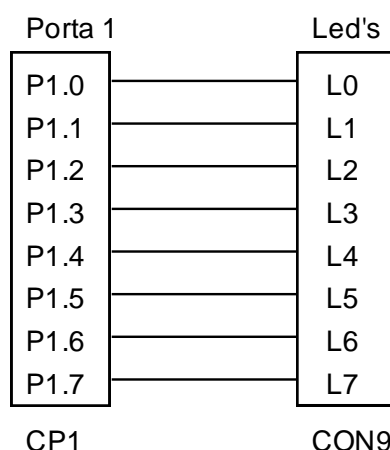
7.19.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

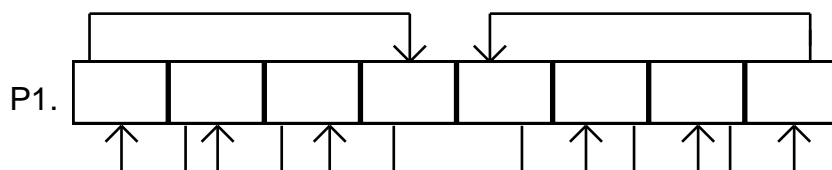
Desktop ou Notebook

7.19.2 PROCEDIMENTO

- Com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

Porta 1	Led's
P1.0	L0
P1.1	L1
P1.2	L2
P1.3	L3
P1.4	L4
P1.5	L5
P1.6	L6
P1.7	L7
CP1	CON9

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o programa seguinte, que visa familiarizar o usuário com as instruções de deslocamento de bits na porta P1.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	78 10	REPETE:	mov R0, # 10h	R0 ← 10h
5002	79 08		mov R1, # 08h	R1 ← 08h
5004	7A 04		mov R2, # 04h	R2 ← 04h
5006	E8	VOLTA:	mov A, R0	Acc ← R0
5007	29		add A, R1	Acc ← R1 + Acc
5008	F5 90		mov P1, A	P1 ← Acc
500A	12 50 1A		lcall ATRASO	busca de atraso
500D	E8		mov A, R0	Acc ← R0
500E	23		rl A	desloca Acc a esquerda
500F	F8		mov R0, A	R0 ← Acc
5010	E9		mov A, R1	Acc ← R1
5011	03		rr A	desloca Acc a direita
5012	F9		mov R1 , A	R1 ← Acc
5013	EA		mov A, R2	Acc ← R2
5014	14		dec A	Acc ← Acc – 1
5015	FA		mov R2 , A	R2 ← Acc
5016	60 E8		jz REPETE	jump se Acc = 0
5018	80 EC		sjmp VOLTA	
501A	C0 00	ATRASO:	push 0	
501C	C0 01		push 1	
501E	C0 02		push 2	
5020	78 03		mov R0,#03h	R0 ← 03
5022	79 FF	SALTO3:	mov R1,#0FFh	R1 ← FF
5024	7A FF	SALTO2:	mov R2, #0FFh	R2 ← FF
5026	DA FE	SALTO1:	djnz R2,SALTO1	
5028	D9 FA		djnz R1,SALTO2	decr.reg. e jump se ≠ 0
502A	D8 F6		djnz R0,SALTO3	
502C	D0 02		pop 2	
502E	D0 01		pop 1	
5030	D0 00		pop 0	
5032	22		ret	

- Executar o programa no modo direto.
- O programa efetua o deslocamento de bits, apresentando o resultado na porta P1.
- Os bits serão deslocados conforme o diagrama a seguir.

7.19.3 PROBLEMA PROPOSTO

- Implementar este mesmo problema usando endereçamento de bit.

7.20 EXPERIÊNCIA 20: SIMULADOR DE PLC COM 6 ENTRADAS E 2 SAÍDAS

Esta experiência mostra a utilização dos pinos da porta **P1** como entradas e saídas de variáveis lógicas, implementando o controle de uma função Booleana.

O microprocessador é um circuito complexo capaz de ser programável para substituir a operação de um circuito. Assim, através de programação pode-se executar as funções lógicas desejadas. Os microcontroladores da família 8051 possuem instruções Booleanas, que facilmente solucionam tais problemas.

Nesta experiência será implementado um circuito que efetua as seguintes operações lógicas:

$$Y1 = A\ B + C$$

$$Y2 = A\ B + D\ E\ F$$

As duas operações lógicas Y1 e Y2, são dependentes de seis entradas (A, B, C, D, E, F). Para uma operação real, equivalente ao desta experiência, os sensores do circuito deveriam ter os seus sinais detectados e condicionados aos níveis lógicos compatíveis com os níveis da porta P1. Por exemplo, a entrada "A" poderia ser um sensor de motor ligado; "B" poderia ser um sensor de limite máximo de corrente, etc.

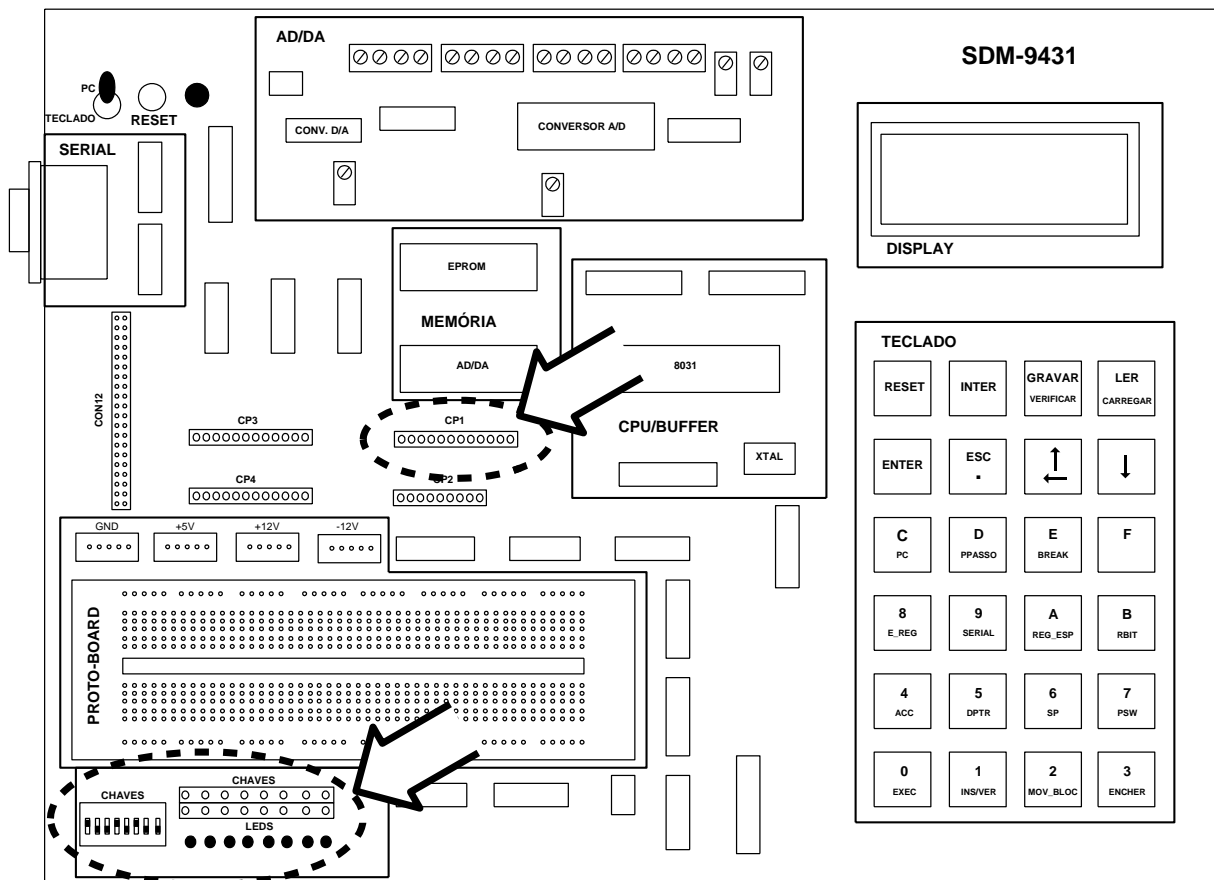
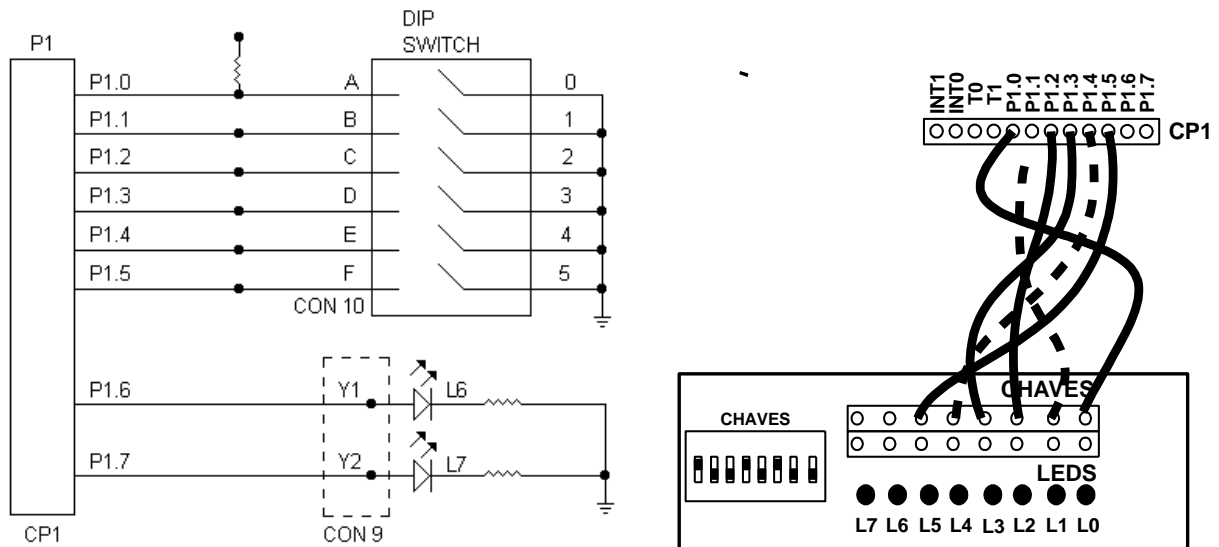
As chaves do módulo SDM-9431 irão simular os valores lógicos dos sensores. Os leds do módulo irão simular as saídas, que poderiam estar ativando um circuito de potência (um motor, um contator, um alarme, etc).

7.20.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.20.2 PROCEDIMENTO

- Com o módulo desligado, conectar os pinos P1.0 até P1.5 às entradas das chaves 0 a 5 do módulo. Conectar o pino P1.6 ao led L6 e o pino P1.7 ao led L7, conforme mostra o esquema seguinte.

- Ligar o módulo e carregar o programa seguinte.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 00		mov A, # 00h	Acc ← 00
5002	F5 20		mov 20h, A	(20h) ← Acc
5004	75 90 3F		mov P1, # 3Fh	P1 ← 3F
5007	A2 91	REPETE:	mov C, 91h	CY ← entrada B
5009	B0 90		anl C, /90h	CY ← B.A
500B	72 92		orl C, 92h	CY ← A.B+C
500D	92 96		mov 96h, C	L6 = Y1← AB+C
500F	A2 90		mov C, 90h	CY ← A
5011	82 91		anl C, 91h	CY ← A.B
5013	92 00		mov 00h, C	armaz. temporária em 20h
5015	A2 93		mov C, 93h	CY ← D
5017	B0 94		anl C, 94h	CY ← D.E
5019	82 95		anl C, 95h	CY ← D.E.F
501B	72 00		orl C, 00h	CY ← AB+DEF
501D	92 97		mov 97h, C	L7 = Y2 ← AB+DEF
501F	80 E6		sjmp REPETE	

- Colocar todas as chaves do módulo posicionadas em zero (para baixo).
- Executar o programa no modo direto.
- Completar as tabelas seguintes.

CHAVE DIP			SAÍDA
0	1	2	L0
A	B	C	Y1
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	0	
1	1	1	
1	0	1	
1	0	0	

CHAVE DIP					SAÍDA
0	1	3	4	5	L1
A	B	D	E	F	Y2
1	1	0	0	0	
1	1	0	0	1	
1	1	0	1	1	
1	1	0	1	0	
1	1	1	1	0	
1	1	1	1	1	
1	1	1	0	1	
1	1	1	0	0	
1	0	1	0	0	
1	0	1	0	1	
1	0	1	1	1	
1	0	1	1	0	
1	0	0	1	0	
1	0	0	1	1	
1	0	0	0	1	
1	0	0	0	0	
0	0	0	0	0	
0	0	0	0	1	
0	0	0	1	1	
0	0	0	1	0	
0	0	1	1	0	
0	0	1	1	1	
0	0	1	0	1	
0	0	1	0	0	
0	1	1	0	0	
0	1	1	0	1	
0	1	1	1	1	
0	1	1	1	0	
0	1	0	1	0	
0	1	0	1	1	
0	1	0	0	1	
0	1	0	0	0	

7.20.3 OBSERVAÇÕES

- A saída L0 = Y1 será 1 somente se a entrada C = DIP2 for 1, ou se A = DIP0 for 0 e B = DIP1 for 1.
- A saída L1 = Y2 será 1 somente se A = DIP0 for 1 e B = DIP1 for 1, ou então se D = DIP3 for 1 e E = DIP4 for 0 e F = DIP5 for 1.
- No programa a instrução MOV P1, # 3Fh foi utilizada para preparar os bits de 0 até 5 da porta P1 como entrada e sair com o nível zero nos bits 6 e 7 de P1. Para se utilizar um bit da porta 1 como entrada é necessário escrever "1", na posição correspondente ao bit, a fim de programar o buffer da porta P1 como entrada.
- O bit 0 do endereço de byte 20h, foi utilizado como memória temporária da equação Y2. O mesmo é acessado pelo endereço de bit 00h.

7.21 EXPERIÊNCIA 21: PROJETO DE UM SISTEMA SEMAFÓRICO

Esta experiência implementa um controle semafórico, operando conforme o seguinte enunciado:

Deseja-se comandar um conjunto de faróis no cruzamento de duas ruas: o farol dirigido para cada uma das ruas, pode estar verde ou vermelho. O carro pode passar se o farol dirigido a ele estiver verde.

Impõe-se as seguintes condições:

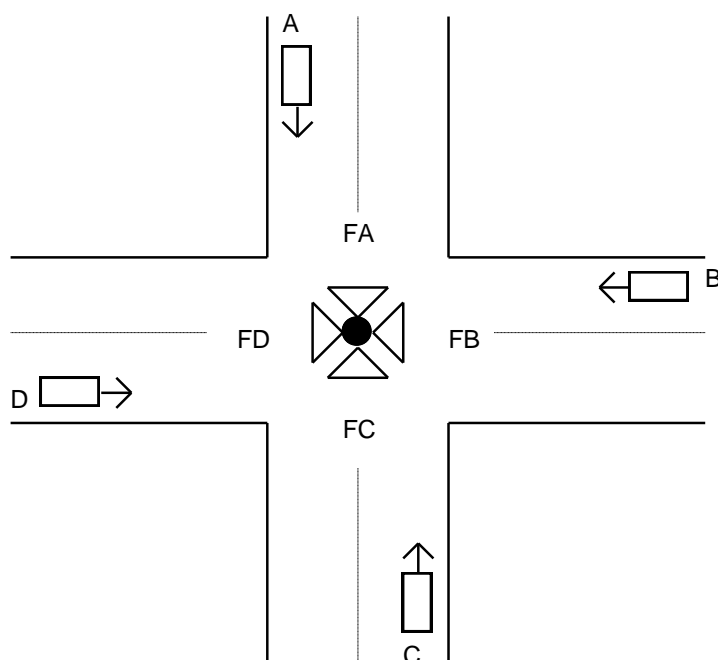
a. Apenas um carro de cada vez deve passar no cruzamento;

b. Se não houver o carro X, o farol FX é um estado opcional;

c. O carro da direita, quando houver, tem a preferência;

d. A ordem de preferência, no caso de haver todos os carros ou no caso dos carros estarem na mesma rua, é sempre ABCD;

e. As condições anteriores são preferenciais na ordem dada.

Projetar o circuito de comando dos faróis, sabendo-se que existe um sistema de fotocélulas para detectar a aproximação dos carros.

São possíveis duas abordagens para solucionar o problema:

a. elaborar um circuito digital que atenda às condições especificadas e

b. elaborar um programa que atenda às condições especificadas.

A. SOLUÇÃO ATRAVÉS DE CIRCUITO LÓGICO

Para obter as expressões booleanas necessárias, adota-se

a. farol verde = 1 (led aceso)

b. farol vermelho = 0 (led apagado)

c. há carro = 1 (chave ligada)

d. não há carro = 0 (chave desligada)

Neste caso há quatro faróis e, consequentemente, quatro saídas. Assim, constrói-se um mapa para cada saída.

FAROL FA = D + BC

FAROL FB = A

FAROL FC = A B + B D

FAROL FD = A C + B C

O raciocínio adotado para o preenchimento dos mapas, pode ser exemplificado pelos exemplos seguintes:

- Seja a posição do mapa onde ABCD = 0101. Isto significa que há carros em B e D e não há carros em A e C. Não havendo carros em A e C, as saídas FA e FC serão opcionais, representadas por "_" nos mapas correspondentes dos faróis FA e FC.
- Quando há carros em B e em D deve-se obedecer às condições do problema. Os carros B e D estão em sentidos opostos da rua. Assim, a condição a ser cumprida será a condição "d", onde o carro B tem a preferência sobre o carro D. Logo, FB = 1 e FD = 0, para este caso.

B. SOLUÇÃO POR PROGRAMAÇÃO

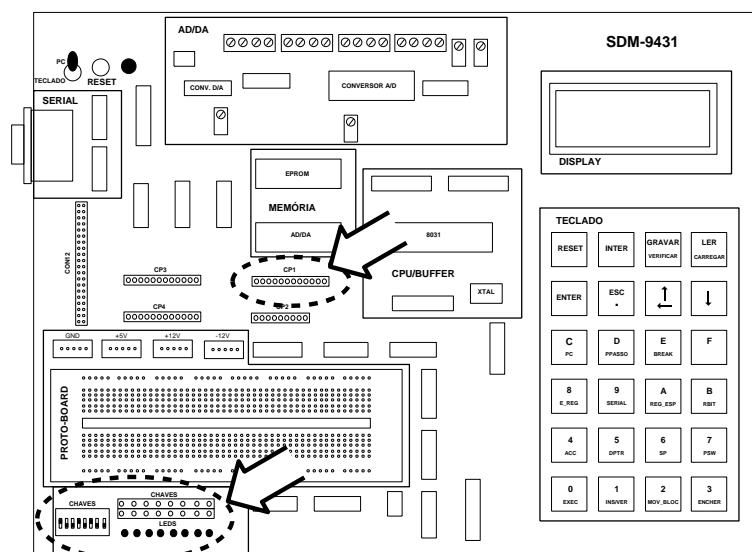
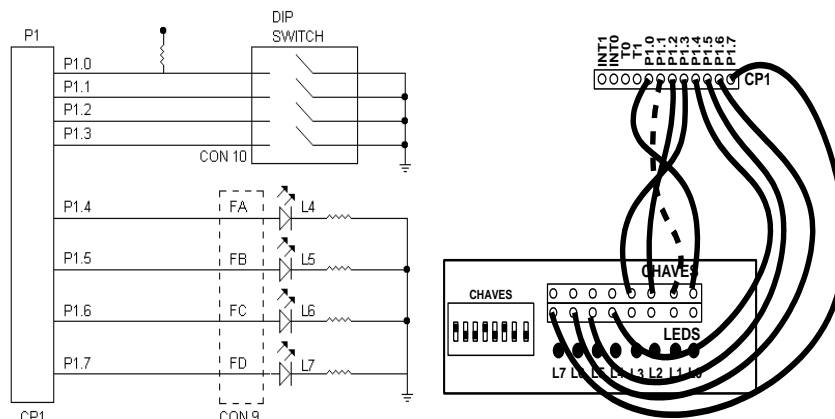
O microcontrolador pode substituir o circuito projetado, através de um programa e realizar a decisão lógica de operação para o projeto em questão.

7.21.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.21.2 PROCEDIMENTO

- Com o módulo desligado, conectar os quatro bits menos significativos da porta P1 às chaves de 0 a 3 e os quatro bits mais significativos da porta P1 aos leds L4 a L7, conforme mostrado a seguir.

- Ligar o módulo e carregar o programa seguinte.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 00		mov A,# 00h	Acc ← 00
5002	F5 20		mov 20h, A	End 20h ← Acc
5004	75 90 0F		mov P1, # 0Fh	P1 ← 0F
5007	A2 91	REPETE:	mov C, 91h	CY ← 91
5009	82 92		anl C, 92h	'E' entre CY e dado End 92
500B	A0 93		orl C, /93h	'OU'entre CY e End 93
500D	92 94		mov 94h, C	End 94 ← CY
500F	A2 90		mov C, 90h	CY ← 90
5011	B3		cpl C	inverte o bit do CY
5012	92 95		mov 95h, C	End 95 ← CY
5014	B0 91		anl C, /91h	'E' entre CY e End 91
5016	92 00		mov 00h, C	End 00 ← CY
5018	A2 93		mov C, 93h	CY ← 93
501A	B0 91		anl C, / 91h	E' entre CY e End 91
501C	72 00		orl C, 00h	'OU'entre CY e End 00
501E	92 96		mov 96h, C	End 96 ← CY
5020	A2 90		mov C, 90h	CY ← 90
5022	B0 92		anl C, /92h	'E' entre CY e End 92
5024	92 01		mov 01h, C	End 01 ← CY
5026	A2 91		mov C, 91h	CY ← 91
5028	B3		cpl C	inverte o bit do CY
5029	B0 92		anl C, /92h	'E' entre CY e End 92
502B	72 01		orl C, 01h	'OU'entre CY e End 01
502D	92 97		mov 97h, C	End 97 ← CY
502F	80 D6		sjmp REPETE	

- Completar os mapas de resultados, sendo led aceso igual a nível lógico 1.

FA = L4

FB = L5

FC = L6

FD = L7

- Comparar os mapas obtidos com os mapas do projeto.

7.21.3 OBSERVAÇÕES

- No projeto em questão os estados opcionais terão as saídas iguais a um, quando forem incluídos na formação do grupo de 1s. Estados opcionais não incluídos nos grupos de 1s, terão as saídas iguais a zero.

7.22 EXPERIÊNCIA 22: PROJETO DE UM GUINDASTE COM LIMITAÇÃO DE CARGAS

Esta experiência simula um guindaste, que deve permitir a elevação de massas compreendidas entre 20 e 80 quilos. Para isso ele comporta uma plataforma, repousando

sobre molas, que possui 3 interruptores (A, B e C), que são acionados pelo peso da carga, respectivamente, com 10kg, 20kg e 80kg.

As condições de funcionamento são as seguintes:

- **a.** A vazio o guindaste deve funcionar;
- **b.** Para cargas entre 10 e 20 quilos o guindaste não deve funcionar;
- **c.** Para cargas compreendidas entre 20 e 80 quilos o guindaste deve operar;
- **d.** Para cargas superiores a 80 quilos o guindaste não pode funcionar.

Deve ser projetado um circuito eletrônico que satisfaça as condições estabelecidas.

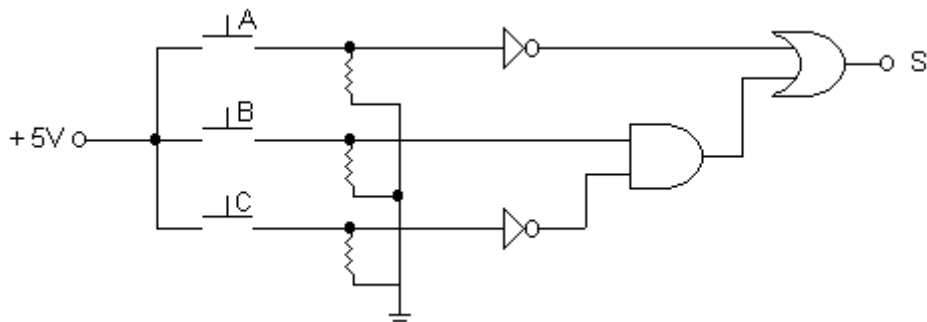
A. <u>SOLUÇÃO ATRAVÉS DE CIRCUITO LÓGICO</u>

A operação do circuito pode ser descrita pela tabela seguinte:

A	B	C	S		CARGA	CONDIÇÃO
0	0	0	**1**		<10kg , <20kg e <80kg	deve operar
0	0	1	**–**		<10kg , <20kg e >80kg	impossível
0	1	1	**–**		<10kg , >20kg e >80kg	impossível
0	1	0	**–**		<10kg , >20kg e <80kg	impossível
1	1	0	**1**		>10kg , >20kg e <80kg	deve operar
1	1	1	**0**		>10kg , >20kg e >80kg	não deve operar
1	0	1	**–**		>10kg , <20kg e >80kg	impossível
1	0	0	**0**		>10kg , <20kg e <80kg	não deve operar

O mapa de Karnaugh correspondente será:

C \ AB	00	01	11	10
0	1	–	1	0
1	–	–	0	_

O circuito lógico correspondente será:

Saídas opcionais podem ser admitidas, desde que haja a certeza da não ocorrência de falhas nos sensores de carga.

B. SOLUÇÃO POR PROGRAMAÇÃO

Um programa poderá substituir o circuito digital, tomando as decisões lógicas correspondentes ao circuito anterior.

7.22.1 EQUIPAMENTO NECESSÁRIO

Módulo SDM 9431

Desktop ou Notebook

7.22.2 PROCEDIMENTO

- Com o módulo desligado, conectar os 3 bits menos significativos da porta P1 às chaves de 0 a 2 e o pino P1.7 ao led L7.
- As chaves DIP 0, 1 e 2 irão simular as entradas dos sensores de carga A, B e C e o led L7 irá simular a saída de controle do guindaste.
- Ligar o módulo e carregar o programa seguinte.

END	OPCODE	LABEL	MNEMÔNICO	COMENTÁRIOS
5000	74 00		mov A,# 00h	Acc ← 00
5002	F5 20		mov 20h, A	End 20h ← Acc
5004	75 90 07		mov P1, # 07h	P1 ← 07
5007	A2 91	REPETE:	mov C, 91h	CY ← 91
5009	B0 92		anl C, /92h	'E' entre CY e dado End 92
500B	A0 90		orl C, /90h	'OU'entre CY e End 90
500D	92 97		mov 97h, C	End 97 ← CY
500F	80 F6		sjmp REPETE	

- Executar o programa no modo direto.
- Completar a tabela seguinte:

CHAVE DIP			SAÍDA
0	1	2	
A	B	C	L7
0	0	0	
0	0	1	
0	1	1	
0	1	0	
1	1	0	
1	1	1	
1	0	1	
1	0	0	

- Compare a tabela obtida com a tabela do projeto.

7.22.3 OBSERVAÇÕES

- Saídas opcionais terão valor lógico um, quando incluídas no grupo de leitura dos 1s e terão valor lógico zero, quando não incluídas.

7.23 EXPERIÊNCIA 23: PROJETO DE UM SISTEMA DE VOTAÇÃO MAJORITÁRIO (PROPOSTO)

Elaborar um circuito digital que atenda às condições descritas a seguir.

Em seguida elaborar um programa para o microcontrolador 8031 que implemente estas condições no módulo SDM-9431.

a. Uma comissão de três pessoas decide por voto majoritário.

b. Cada membro pode pressionar um botão para significar um voto "sim".

c. Elaborar um circuito lógico com blocos não-E que acenda uma lâmpada quando, e somente quando, a maioria dos votos for "sim".

d. Resolva o problema, considerando que os votos "sim" corresponde ao nível lógico um e o voto "não" corresponde ao nível lógico zero.

7.24 EXPERIÊNCIA 24: DESENVOLVIMENTO DE PROGRAMAS USANDO A LINGUAGEM ASSEMBLY

Vários programas de desenvolvimento de sistemas para a família de microcontroladores 8051, incluindo compiladores e simuladores, são disponíveis no mercado.

O sistema SDM 9431 reconhece os arquivos obtidos no padrão **.HEX**, gerados através destes programas. A seguir será exemplificada a sequência de operações necessárias à obtenção de um programa **.HEX** com o uso do **AVCASE**, da Avocet Systems.

- Inicialmente, usando um editor de texto qualquer, deve-se criar um arquivo texto no formato **.ASM**, obedecendo todas as regras necessárias para a linguagem mnemônica. Deve-se consultar os manuais dos fabricantes para obter tais informações.
- Todo programa deverá iniciar com a seguinte declaração:

  $R0(0) - R0 (0000h - 1FFFh) - XR (2000h - 0FFFFh) - Nh

  que definirá a área de blocos de memória para a ROM e RAM do sistema. A opção **-Nh** define sobreposição da ROM e RAM.
- Após a edição do arquivo **.ASM** deve ser usado o programa **AVA51**, para efetuar o assembler do arquivo texto, criando os arquivos **.OBJ** e **.PRN**.
- Se a operação possuir erros, os mesmos estarão indicados no arquivo **.PRN**.
- A execução do "ASSEMBLER" será feita através do comando:

  AVA 51 nome do_arquivo

- Após a execução do assembler deve ser executado o programa **AVL51**, que irá realizar as ligações (LINKER) entre os programas objetos necessários e irá gerar os arquivos **.HEX**, **.MAP**, **.SYM** e **.SMB**.
- A execução do "LINKER" será feita através do comando:

AVL51 nome = nome_do_arquivo.OBJ, -SY -DB

- O arquivo **.HEX** criado será usado para comunicação com o sistema SDM 9431, através da porta serial e do comando carregar arquivos disponível no módulo, tanto no modo PC, quanto no modo Teclado.
- Os programas correspondentes às experiências apresentadas neste capítulo estão disponíveis nos formatos **.ASM**, **.HEX** e **.PRN** na documentação que acompanha o sistema SMD 9431.

7.25 EXPERIÊNCIA 25: DESENVOLVIMENTO DE PROGRAMAS USANDO LINGUAGEM C

O uso de linguagem de alto nível possibilita uma melhor organização no desenvolvimento de programas, devido à utilização de palavras ou linhas de comandos, que melhor definem a tarefa a ser executada.

Para a família 8051 é empregada a programação em linguagem C, que sendo uma uma linguagem de alto nível, resulta em uma melhor documentação do programa.

Esta linguagem possibilita um controle direto do sistema, com um grau de liberdade de uso de apontadores e endereçamentos indiretos para referenciar a memória, com o código fonte sendo facilmente transportado para outras máquinas.

Todas as operações de baixo nível, tais como entradas e saídas poderão ser realizadas através de funções de uma biblioteca. Ainda, esta linguagem permite utilizar trechos do programa escrito em linguagem assembly, quando se deseja executar funções específicas da máquina, no menor tempo possível.

A desvantagem de se utilizar o desenvolvimento em linguagem de alto nível está na redução da eficiência do programa, visto que a compilação do mesmo, ou seja a tradução da linguagem C para a linguagem de máquina, resultará em um programa que usa maior quantidade de memória, se comparado a um programa escrito em linguagem assembly. Isto implicará em um programa que será executado em um tempo maior.

Para utilização da linguagem C com o sistema SDM 9431, pode-se usar o programa AVCASE, que possui um compilador C.

- Para tanto, deve-se escrever um arquivo de configuração, definindo a área de memória disponível para o sistema, conforme o exemplo do arquivo AVC51.CFG apresentado na figura seguinte. Para o melhor entendimento deste comando, deve ser consultado o manual do compilador C do AVCASE.

```
- model: mc
- Verbose: 0
- RAM: 0 - FF
- ROM: 5000 - 5FFF
- XRAM: 5000 - 5FFF
- WARNINGLEVEL: 10
- OPT JMP
- STDINC: c\avc51\include
- asmlist
```

- Após a edição do arquivo .C deve ser usado o programa AVC51.EXE, para efetuar a compilação do arquivo texto. Isto é feito através do comando

  AVC51 nome-do-arquivo.C

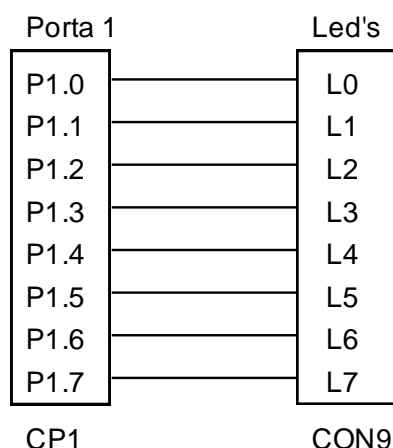
  que criará os arquivos **.PRN** e **.HEX**, se a operação estiver isenta de erros.
- O arquivo **.HEX** será transferido para o sistema SDM 9431 através de comunicação serial e poderá ser executado.
- A seguir tem-se um exemplo de um programa escrito em linguagem C.

```
;################################################################
; Programa exemplo de utilizacao de linguagem C
; para o modulo SDM9431.
; Este programa, utiliza uma estrutura para enderecamento
; de bit da porta P1 e uma variavel para enderecamento de
; byte da porta P1.
;################################################################

#define porta (*(struct porta1 *) 0x90)
#define outport (*(char *) 0x90)
char valor;
struct porta1
        {
         unsigned bit0:1,
                  bit1:1,
                  bit2:1,
                  bit3:1,
                  bit4:1,
                  bit5:1,
                  bit6:1,
                  bit7:1;
```

```
          };

void outbyte (char valor)
          {
           outport = valor;
          }

           int a,b,c;
void atraso()
           {
           for (a=0; a < 20;a++)
                    {
                     for (b=0;b < 25;b++)
                              {
                               for (c=0;c < 150;c++)
                                        {

                                        }
                              }
                    }
           }


void main()

     {
          while(1)
            {
             porta.bit0=0;
             porta.bit1=1;
             porta.bit2=0;
             porta.bit3=1;
             porta.bit4=1;
             porta.bit5=0;
             porta.bit6=1;
             porta.bit7=0;
             atraso();
             valor=0xc5;
             outbyte(valor);
             atraso();
            }
    }
```

- Este programa cria uma estrutura denominada "porta" para o manuseio dos bits endereçáveis da porta P1 e define uma variável "outport" para manuseio de byte da porta P1.
- Inicializa os bits da porta P1 com:

bit 0 = 0	*bit 4 = 1*
bit 1 = 1	*bit 5 = 0*
bit 2 = 0	*bit 6 = 1*
bit 3 = 1	*bit 7 = 0*

- Executa uma função de atraso.
- Inicializa a variável "valor" com o código hexadecimal C5h_ e envia esta variável para a porta P1, chamando novamente a função de atraso.
- Assim, é ciclicamente alterado o conteúdo da porta P1.
- Na documentação fornecida estão apresentados os arquivos EXP25.C, EXP25.HEX e EXP25.PRN, correspondentes a esta experiência.
- Para testar esta execução, com o equipamento desligado, fazer a conexão entre os bits da porta P1, denominados de P1.0 até P1.7, no barramento CP1 e conjunto de leds denominados L0 até L7, conforme o esquema seguinte.

- Ligar o módulo retornando a operação no modo Teclado, ou no modo PC.
- Carregar o arquivo EXP25.HEX.
- Executar no modo direto.
- A seguir tem-se a listagem do arquivo EXP25.PRN gerada pelo compilador C.

```
;##########################################
; Arquivo .PRN para experiencia 25 gerado
; pelo compilador C do AVCASE51.
;##########################################

                    1    ;teste.c: 3: char valor;
                    2    ;teste.c: 4: struct porta1
                    3    ;teste.c: 5:        {
                    4    ;teste.c: 6:         unsigned bit0:1,
                    5    ;teste.c: 7:                  bit1:1,
                    6    ;teste.c: 8:                  bit2:1,
                    7    ;teste.c: 9:                  bit3:1,
                    8    ;teste.c: 10:                 bit4:1,
                    9    ;teste.c: 11:                 bit5:1,
                    10   ;teste.c: 12:                 bit6:1,
                    11   ;teste.c: 13:                 bit7:1;
                    12   ;teste.c: 14:       };
                    13   ;teste.c: 16: void outbyte (char valor)
                    14   ;teste.c: 17:       {
                    15           global  stack_internal
                    16           defseg  c_text,class=CODE
                    17           global  _outbyte
  =1038                 18       signat  _outbyte,4152
                    19           seg     c_text
0000                20   _outbyte:
0000 8D 90              21       mov     p1,r5
                    22   ;teste.c: 19:       }
0002 22             23           ret
                    24           global  _atraso
  =0018                 25       signat  _atraso,24
                    26           global  _a
                    27           global  _b
                    28           global  _c
                    29   ;teste.c: 21:        int a,b,c;
                    30   ;teste.c: 22: void atraso()
                    31   ;teste.c: 23:        {
0003                32   _atraso:
0003 90 0001!       33           mov     dptr,#_a
0006 E4             34           clr     a
0007 F0             35           movx    @dptr,a
0008 A3             36           inc     dptr
0009 80 6C'             37       jmp     A1
```

```
000B            38    l8:
000B 90 0005    39          mov     dptr,#_c
000E E4         40          clr     a
000F F0         41          movx    @dptr,a
0010 A3         42          inc     dptr
0011 80 14'           43            jmp     A3
0013            44    l12:
0013 90 0005    45          mov     dptr,#_c
0016 E0         46          movx    a,@dptr
0017 FC         47          mov     r4,a
0018 A3         48          inc     dptr
0019 E0         49          movx    a,@dptr
001A 24 01            50            add     a,#1
001C FD               51            mov     r5,a
001D EC               52            mov     a,r4
001E 34 00            53            addc    a,#0
0020 FC         54          mov     r4,a
0021 ED         55          mov     a,r5
0022 F0         56          movx    @dptr,a
0023 90 0005    57          mov     dptr,#_c
0026 EC         58          mov     a,r4
0027            59    A3:
0027 F0         60          movx    @dptr,a
0028 90 0005!   61          mov     dptr,#_c
002B E0         62          movx    a,@dptr
002C FC               63            mov     r4,a
002D A3         64          inc     dptr
002E E0         65          movx    a,@dptr
002F FD         66          mov     r5,a
0030 24 6A            67            add     a,#106
0032 EC         68          mov     a,r4
0033 34 FF            69            addc    a,#255
0035 20 D2# 54' 70          bb      ov,u962
0038 20 E7# D8' 71          bb      acc.7,l12
003B            72    l10:
003B 90 0003!   73          mov     dptr,#_b
003E E0         74          movx    a,@dptr
003F FC         75          mov     r4,a
0040 A3         76          inc     dptr
0041 E0         77          movx    a,@dptr
0042 24 01            78            add     a,#1
0044 FD         79          mov     r5,a
```

```
0045 EC          80          mov     a,r4
0046 34 00          81           addc    a,#0
0048 FC          82          mov     r4,a
0049 ED          83          mov     a,r5
004A F0          84          movx    @dptr,a
004B 90 0003!    85          mov     dptr,#_b
004E EC             86           mov     a,r4
004F             87    A2:
004F F0          88          movx    @dptr,a
0050 90 0003!    89          mov     dptr,#_b
0053 E0          90          movx    a,@dptr
0054 FC          91          mov     r4,a
0055 A3          92          inc     dptr
0056 E0          93          movx    a,@dptr
0057 FD          94          mov     r5,a
0058 24 E7          95           add     a,#231
005A EC             96           mov     a,r4
005B 34 FF          97           addc    a,#255
005D 20 D2# 31'  98          bb      ov,u1152
0060 20 E7# A8'  99          bb      acc.7,l8
0063             100   l6:
0063 90 0001!    101         mov     dptr,#_a
0066 E0          102         movx    a,@dptr
0067 FC          103         mov     r4,a
0068 A3          104         inc     dptr
0069 E0          105         movx    a,@dptr
006A 24 01          106          add     a,#1
006C FD             107          mov     r5,a
006D EC             108          mov     a,r4
006E 34 00          109          addc    a,#0
0070 FC          110         mov     r4,a
0071 ED          111         mov     a,r5
0072 F0          112         movx    @dptr,a
0073 90 0001!    113         mov     dptr,#_a
0076 EC          114         mov     a,r4
0077             115   A1:
0077 F0          116         movx    @dptr,a
0078 90 0001!    117         mov     dptr,#_a
007B E0          118         movx    a,@dptr
007C FC             119          mov     r4,a
007D A3          120         inc     dptr
007E E0          121         movx    a,@dptr
```

```
007F FD          122          mov     r5,a
0080 24 EC            123           add     a,#236
0082 EC          124          mov     a,r4
0083 34 FF            125           addc    a,#255
0085 20 D2# 11'  126          bb      ov,u1342
0088 20 E7# 12'  127          bb      acc.7,l4
008B 22          128          ret
008C             129    u962:
008C 30 E7# 84'  130          bnb     acc.7,l12
008F 80 AA'           131           jmp     l10
0091             132    u1152:
0091 20 E7# 03 02     133           bnb     acc.7,l8
0095 000B'
0097 80 CA'           134           jmp     l6
0099             135    u1342:
0099 30 E7# 01'  136          bnb     acc.7,l4
009C 22          137          ret
009D             138    l4:
009D 90 0003!    139          mov     dptr,#_b
00A0 E4          140          clr     a
00A1 F0          141          movx    @dptr,a
00A2 A3          142          inc     dptr
00A3 80 AA'           143           jmp     A2
                 144          global  _main
   =0018              145           signat  _main,24
                 146          global  _valor
                 147    ;teste.c: 36: void main()
                 148    ;teste.c: 38:     {
00A5             149    _main:
00A5             150    l18:
00A5 C2 97            151           clr     p1.7
                 152    ;teste.c: 42:        (*(struct porta1 *) 0x90)
                                    .bit1=1;
00A7 D2 96            153           setb    p1.6
                 154    ;teste.c: 43:        (*(struct porta1 *) 0x90)
                                    .bit2=0;
00A9 C2 95            155           clr     p1.5
                 156    ;teste.c: 44:        (*(struct porta1 *) 0x90)
                                    .bit3=1;
00AB D2 94            157           setb    p1.4
                 158    ;teste.c: 45:        (*(struct porta1 *) 0x90)
                                    .bit4=1;
```

```
00AD D2 93             159              setb    p1.3
               160     ;teste.c: 46:      (*(struct porta1 *) 0x90)
                                        .bit5=0;
00AF C2 92             161              clr     p1.2
               162     ;teste.c: 47:      (*(struct porta1 *) 0x90)
                                        .bit6=1;
00B1 D2 91             163              setb    p1.1
               164     ;teste.c: 48:      (*(struct porta1 *) 0x90)
                                        .bit7=0;
00B3 C2 90             165              clr     p1.0
               166     ;teste.c: 49:      atraso();
00B5 12 0003'  167              lcall   _atraso
               168     ;teste.c: 50:      valor=0xc5;
00B8 90 0000!  169              mov     dptr,#_valor
00BB 74 C5             170              mov     a,#-59
00BD F0        171              movx    @dptr,a
               172     ;teste.c: 51:      outbyte(valor);
00BE 90 0000!  173              mov     dptr,#_valor
00C1 E0        174              movx    a,@dptr
00C2 FD                175              mov     r5,a
00C3 12 0000'  176              lcall   _outbyte
               177     ;teste.c: 52:      atraso();
00C6 12 0003'  178              lcall   _atraso
               179     ;teste.c: 39:      }
00C9 80 DA'            180              jmp     l18
               181              defseg  c_bss,class=XDATA
               182              seg     c_bss
0000           183     _valor:
0000 (0001)            184              ds      1
0001           185     _a:
0001 (0002)            186              ds      2
0003           187     _b:
0003 (0002)            188              ds      2
0005           189     _c:
0005 (0002)            190              ds      2
               191              end
```

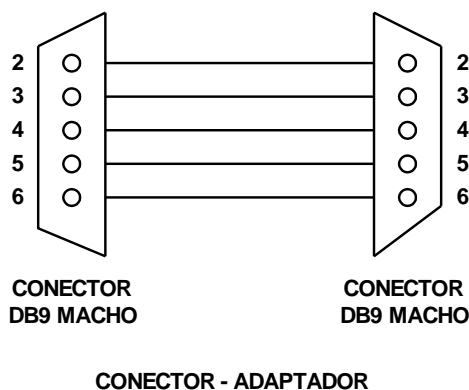
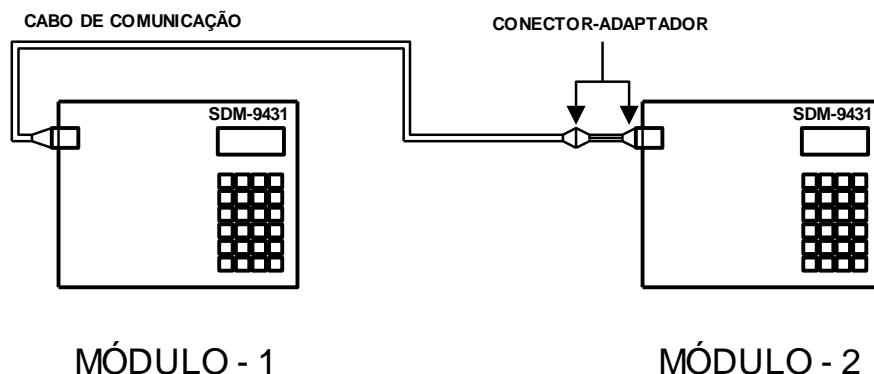
7.26 EXPERIÊNCIA 26: COMUNICAÇÃO SERIAL

7.26.1 EQUIPAMENTO NECESSÁRIO

2 Módulos SDM 9431

Desktop ou Notebook

7.26.2 PROCEDIMENTO

- Carregar o programa *EXP26.HEX* nos módulos 1 e 2. (Esta operação poderá usar o modo PC de programação).
- Desconectar o cabo serial de comunicação do PC e ligá-lo no conector-adaptador, cujo diagrama de ligações encontra-se na figura abaixo:

- Ligar as saídas seriais dos módulos SDM9431 entre si, com o cabo conector-adaptador, obtendo a ligação seguinte:

MÓDULO - 1 MÓDULO - 2

- No modo Teclado, executar o programa carregado nos módulos 1 e 2, a partir do endereço 5000h;
- Pressionar uma tecla, no teclado hexadecimal do módulo 1 (exceto as teclas RESET e INT) e observar os displays dos módulos;
- Pressionar uma tecla, no teclado hexadecimal do módulo 2 e observar, também, os displays dos módulos;

7.26.3 OBSERVAÇÕES

Quando é pressionada uma tecla em qualquer um dos módulos, o código da mesma é mostrado no campo “TX=”, indicando que este módulo está operando como transmissor serial. No campo “RX=” do segundo módulo irá aparecer o valor da tecla pressionada, indicando que o segundo módulo operou como receptor serial. A comunicação entre os módulos é bidirecional.

Comunicação entre um módulo SDM-9431 e o hiperterminal (PC)

- Ligar o cabo de comunicação serial entre um módulo SDM9431 e o PC;
- Carregar, no modo PC o arquivo *EXP26.HEX*;
- Encerrar o programa SDM9431.EXE;
- Executar o programa *Hyperterminal* do Windows (se o PC que está sendo utilizado não possuir o software, copiá-lo do CD de instalação do SDM9431);
- No modo teclado, executar o programa no módulo, a partir do endereço 5000h
- Pressionar qualquer tecla no teclado do PC e observar o display do módulo;
- Pressione uma tecla no teclado do módulo (exceto RESET e INTER) e observar a tela do programa Hyperterminal.

7.26.4 PROBLEMA PROPOSTO

Ao pressionar uma tecla no PC, o programa Hyperterminal mostra-a no monitor e também envia o código da mesma via porta serial. O módulo recebe o dado e mostra-o no display.

Ao pressionar qualquer tecla no teclado do módulo (exceto RESET e INTER), o mesmo além de mostrar o dado no display, também envia-o via porta serial. O PC recebe o dado e mostra o código referente ao dado recebido. O Hyperterminal interpreta o código como sendo ASCII e também envia o dado nesse formato.

O fluxograma e o programa detalhado de funcionamento da experiência encontram-se a seguir:

INICIO

INICIALIZAÇÃO DO STACK

INICIALIZAÇÃO DA PORTA SERIAL 1 START, 8 BITS, 2 STOP TAXA DE 9600 BPS

LIMPA O DISPLAY E ESCREVE AS MENSSAGENS "RX" E "TX"

HABILITA INTERRUPÇÃO DA PORTA SERIAL

OCORREU RECEBIMENTO DE DADO NA SERIAL?

SIM

GERA INTERRUPÇÃO SERIAL DE RX

NÃO

1

LÊ TECLADO

NÃO

ALGUMA TECLA FOI PRESSIONADA?

SIM

ENVIA O DADO PELA SERIAL

OCORREU INTERRUPÇÃO SERIAL DO TRANSMISSOR?

NÃO

SIM

GERA INTERRUPÇÃO SERIAL DE TX

2

INTERRUPÇÃO

DESABILITA INTERRUPÇÃO

SIM

É INTERRUPÇÃO RX?

DADO RECEBIDO É MOSTRADO NO DISPLAY

ZERA RI E HABILITA INTERRUPÇÃO

RETORNO DE INTERRUPÇÃO 1

NÃO

SIM

É INTERRUPÇÃO TX?

NÃO

RETORNO DE INTERRUPÇÃO

DADO TRANSMITIDO É MOSTRADO NO DISPLAY

ZERA TI E HABILITA INTERRUPÇÃO

RETORNO DE INTERRUPÇÃO 2

```
1:                          ;=================================
 2:                         ; DATAPOOL ELETRÔNICA LTDA
 3:                         ;
 4:                         ; PROGRAMA DE COMUNICAÇÃO SERIAL
 5:                         ; EXP26.H
 6:                         ;
 7:                         ;=================================
 8:
 9:
10:                         ;ROTINAS DO PROGRAMA MONITOR
11:
12:    N    10AA     clr_dsp        equ     10AAh
13:    N    1002     le_tec         equ     1002h
14:    N    0F00     le_dado        equ     0F00h
15:    N    109A     dsp_com        equ     109Ah
16:    N    10FF     dsp_dat        equ     10FFh
17:    N    10E7     ac_dsp         equ     10E7h
18:    N    110F     mens            equ    110Fh
19:
20:
21:                      ;TRATAMENTO DE INTERRUPÇÃO
22:  N     4270          org      4270h
23:  4270  02 50 42      ljmp     inter_serial       ;VAI PARA A ROTINA DE TRATAMENTO
24:                                                  ;DE INTERRUPÇÃO.
25:
26:
27:                      ;PROGRAMA PRINCIPAL
28:   N     5000          org      5000h
29:
30:  5000  75 81 2F       mov      sp,#2Fh            ;INICIALIZA O STACK POINTER.
31:
32:  5003  12 50 2F       lcall    inicia_serial      ;CONFIGURA A SERIAL.
33:
34:  5006  12 10 AA       lcall    clr_dsp            ;LIMPA O DISPLAY.
```

```
35:
36:  5009  90 50 69          mov     dptr,#men_tx       ;ESCREVE A MENSSAGEM
37:  500C  12 11 0F          lcall   mens               ;"TX=" NO DISPLAY.
38:
39:  500F  74 C0             mov     A,#0c0h            ;POSICIONA O CURSOR DO
40:  5011  12 10 9A          lcall   dsp_com            ;DISPLAY NA 2º LINHA E
41:  5014  90 50 6D          mov     dptr,#men_rx       ;ESCREVE A MENSSAGEM
42:  5017  12 11 0F          lcall   mens               ;"RX=" NO DISPLAY.
43:
44:  501A  74 0F             mov     A,#0Fh             ;MUDA A CINTILAÇÃO
45:  501C  12 10 9A          lcall   dsp_com            ;DO DISPLAY.
46:
47:  501F  75 A8 90          mov     IE,#10010000B      ;HABILITA INTERRUPÇÃO DA PORTA SERIAL.
48:
49:  5022  74 83   loop:     mov     A,#83h             ;POSICIONA O CURSOR NA 1ª LINHA
50:  5024  12 10 9A          lcall   dsp_com            ;4ª COLUNA.
51:  5027  12 10 02          lcall   le_tec             ;CHAMA A ROTINA DE LEITURA DO TECLADO.
52:  502A  FD                MOV     R5,A               ;GUARDA O VALOR DA TECLA PRESSIONANDA
53:                                                     ;NO REGISTRO R5.
54:  502B  8D 99             MOV     SBUF,R5            ;ENVIA O DADO PELA SERIAL.
55:  502D  80 F3             sjmp    loop               ;INICIA NOVO CICLO.
56:
57:                               ;INICIALIZAÇÃO DA SERIAL
58:
59:  502F  75 89 20 inicia_serial: mov   TMOD,#20h        ;TC/1 :temporizador de 8 bits c/ recarga automatica.
60:  5032  75 8D F9                mov   TH1,#249         ;valor a recarregar (taxa = 8928,5714).
61:  5035  75 8B F9                mov   TL1,#249
62:  5038  75 98 D8                mov   SCON,#11011000B  ;Modo 3 de transm.( 1 start bit, 8 dados, 2 stop bits).
63:  503B  75 87 80                mov   PCON,#80h        ;Dobra a taxa de tansm. (SMOD=1).
64:  503E  75 88 C0                mov   TCON,#0C0h       ;Liga TC/1 (TR1=1).
65:  5041  22                      ret
66:
67:                               ;TRATAMENTO DA INTERRUPÇÃO
68:
69:  5042  75 A8 00 inter_serial:  mov   IE,#00h          ;Desabilita todas interrupções.
```

```
70:
71:  5045  E5 98              MOV   A,SCON          ;VERIFICA SE
72:  5047  54 01              ANL    A,#01h                  ;O DADO FOI RECEBIDO
73:  5049  B4 01 0E           CJNE  A,#01h                  ,TST_TX  ;OU TRANSMITIDO.
74:
75:                           ;SE O DADO FOI RECEBIDO
76:  504C  74 C3              MOV      A,#0C3h      ;POSICIONA O CURSOR
77:  504E  12 10 9A           LCALL    DSP_COM      ;NA 2ª LINHA 4ª COLUNA.
78:  5051  E5 99              MOV      A,SBUF       ;TRANSFERE O DADO PARA O ACUMULADOR E
79:  5053  12 10 E7           LCALL    AC_DSP       ;MOSTRA-O NO DISPLAY.
80:  5056  C2 98              clr      RI           ;ZERA O BIT RI DO REGISTRO SCON
81:  5058  80 0B              SJMP     FIM_INT      ;E VOLTA AO PROGRAMA PRINCIPAL.
82:
83:                           ;SE O DADO FOR TRANSMITIDO
84:  505A  74 83  TST_TX:     MOV     A,#083h       ;POSICIONA O CURSOR
85:  505C  12 10 9A           LCALL   DSP_COM       ;NA 1ª LINHA 4ª COLUNA.
86:  505F  ED                 MOV     a,R5          ;TRANSFERE O DADO DO REGISTRO PARA O ACUMULADOR
87:  5060  12 10 E7           LCALL   AC_DSP        ;E MOSTRA-O NO DISPLAY.
88:  5063  C2 99              clr     TI            ;ZERA-SE O BIT TI DO REGISTRO SCON.
89:
90:  5065  75 A8 90 FIM_INT:   mov    IE,#10010000B          ;HABILITA INTERRUPÇÕES
91:  5068  32                 reti                  ;RETORNA DA ROTINA DE TRATAMENTO DE INTERRUPÇÃO.
92:
93:  5069  03 54 58 3D    men_tx:  db  3,'TX='      ;MENSAGEM "TX="
94:  506D  03 52 58 3D    men_rx:  db  3,'RX='      ;MENSAGEM "RX="
95:
96:                           end                   ;FIM DO PROGRAMA
```

7.27 EXPERIÊNCIA 27: DIGITALIZAÇÃO DE VOZ

Nesta experiência, utilizaremos o conversor Analógico-Digital (AD) e o Conversor Digital-Analógico (DA) para fazer a digitalização de um sinal de áudio vindo de um microfone.

7.27.1 CARACTERÍSTICAS DA PLACA CIP0931

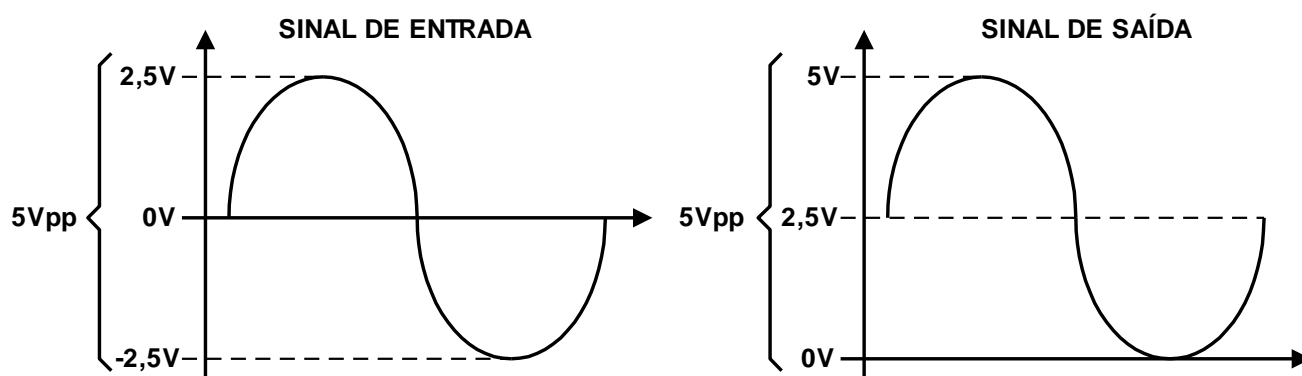
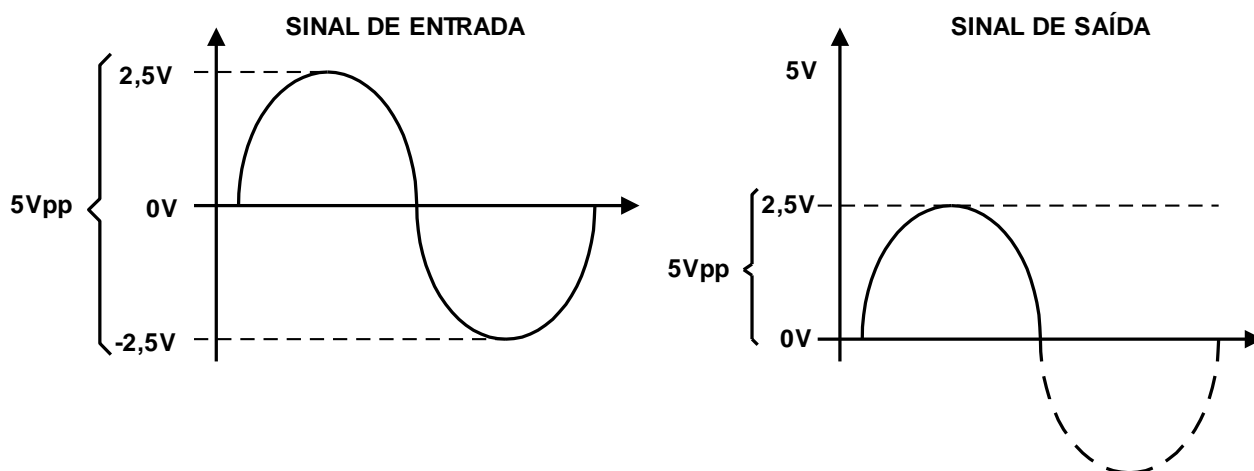
Para esta experiência usaremos a placa **CIP0931 (opcional)** que possui as seguintes características:

- Um pré amplificador para microfone;
- Uma saída de áudio ligada a um alto-falante;
- Um circuito passa-baixa;
- Um amplificador que amplifica sinais até 0,1V;
- Um amplificador que amplifica sinais até 1V.

7.27.2 CONSIDERAÇÕES IMPORTANTES

- **CONVERSOR AD**

O nível de tensão máxima na entrada do conversor AD é de 0 a 5V. Mas o sinal de áudio é um sinal alternado. Logo devemos fazer uma "compensação" somando um sinal de OFFSET para trabalharmos dentro da faixa de 0 a 5V. Esta tensão é de 2,5V.

Nas figuras a seguir, podemos analisar melhor o efeito desta "compensação".

Sem esta "compensação", o conversor só trabalharia com o sinal que estiver acima de 0V.

- **CONVERSOR DA**

No módulo SDM-9431 possui o jumper JP4 localizado no canto superior esquerdo. Este jumper seleciona o nível de saída do conversor DA. Quando conectado entre os terminais de cima e do meio, na saída do conversor DA o sinal estará entre –5 a +5V. Quando o mesmo estiver conectado entre o terminal debaixo e do centro a saída estará entre 0 a +5V.

Nesta experiência utilizaremos a saída entre –5V a +5V, logo o jumper JP4 deverá estar entre os terminais de cima e do meio.

7.27.3 EQUIPAMENTO NECESSÁRIO

Placa CIP 0931
(opcional)

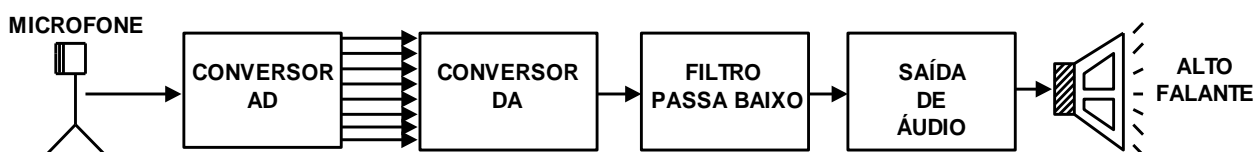
7.27.4 PROCEDIMENTO

- Desligue o módulo SDM-9431;
- No módulo SDM-9431 coloque o jumper JP4 localizado no canto superior esquerdo entre os terminais de cima e do meio;
- Conecte a placa CIP0931 nos terminais do protoboard do módulo;
- Ligue os conectores de alimentação da placa CIP0931 (CN1), nos conectores de alimentação do módulo SDM-9431 (+5V, +12V e GND);
- Conecte o microfone na placa CIP0931;
- Ligue o pino SMIC do conector CN3 da placa CIP0931 na Entrada EA1 do módulo SDM-9431 (Conector CON4);
- Ligue o pino AF do conector CN3 da placa CIP0931 na Saída DAC do módulo SDM-9431 (Conector CON6);
- Confira as ligações feitas (analisando novamente os passos anteriores), se tudo estiver certo, ligue o módulo;
- Estando o módulo no modo PC ou no módulo teclado, carregue a experiência 27;
- Execute o programa;
- Fale próximo ao microfone e perceba que o mesmo sinal na entrada do microfone é o mesmo na saída do alto falante.

7.27.5 OBSERVAÇÕES

O Sinal de voz passa pelo microfone e em seguida segue para o pré-amplificador que amplifica e corrige o sinal somando uma tensão de 2,5V (Este sinal está limitado entre 0 a 5V).

Em seguida o sinal é injetado na entrada EA1 do conversor AD que o digitaliza. Logo em seguida, o sinal digital é injetado no conversor DA que o “transforma” novamente em sinal analógico. Este sinal analógico novamente passa por um filtro “passa-baixa” para retirar sinais indesejáveis e em seguida o mesmo segue para a etapa de áudio para ser amplificado e jogado em um alto falante.

O diagrama em blocos da experiência se encontra na figura a seguir:

O programa para o funcionamento da experiência se encontra a seguir:

```
 1:                          ;================================
 2:                          ; DATAPOOL ELETRONICA
 3:                          ; PROGRAMA PARA UTILIZAÇÃO DOS
 4:                          ; CONVERSORES AD E DA
 5:                          ;================================
 6:
 7:
 8:                          ;ENDEREÇO DAS ROTINAS USADAS
 9:
10:        N    1471         da1     equ   1471h
11:        N    145F         ad      equ   145Fh
12:        N    E000         EA1     equ   0E000h
13:
14:
15:                                  ;INICIO DO PROGRAMA
16:        N    5000                 org    5000h
17:
18:  5000 75 81 2F                   mov    sp,#2Fh      ;CARREGA O STACK COM 2Fh
19:  5003 90 E0 00      repete:      mov    dptr,#EA1    ;DPTR APONTA PARA O EA1
20:  5006 12 14 5F                   lcall  ad           ;CHAMADA DA ROTINA AD
21:  5009 12 50 11                   lcall  atraso       ;CHAMADA DA ROTINA ATRASO
22:  500C 12 14 71                   lcall  da1          ;CHAMADA DA ROTINA DA1
23:  500F 80 F2                      sjmp   repete       ;REPETE TUDO NOVAMENTE
24:
25:
26:                                  ;ROTINA DE ATRASO
27:  5011 C0 00         atraso:            push   0
28:  5013 C0 01                      push   1
29:  5015 C0 02                      push   2
```

```
30:  5017 78 01                 mov   r0,#01h
31:  5019 79 01      salto3:    mov   r1,#01h
32:  501B 7A FF      salto2:    mov   r2,#0FFh
33:  501D DA FE      salto1:    djnz  r2,salto1
34:  501F D9 FA                 djnz  r1,salto2
35:  5021 D8 F6                 djnz  r0,salto3
36:  5023 D0 02                 pop   2
37:  5025 D0 01                 pop   1
38:  5027 D0 00                 pop   0
39:  5029 22                    ret
40:
41:                             end
```